REV.	BY & DATE	DESCRIPTION	CHECK	APPROVED & DATE
A	PLM/JEE 11/20/2002	Corrected Size of GRH. Added Byte sizes to Bit sizes in tables. Corrected Time Format to Jan 1, 1958 epoch. Added SSH information. Removed all DSCM references. Updated Telemetry Directives. Some are obsolete. Some are new. Time partitions now needed for playback. Added Conventions Appendix. Added User Guide to Telemetry Retrieval Appendix. Added requirement for Back-up POC Command Acceptor. Added detail to ApID table. Added details to all tables. Updated time playback data will be available. Removed the Revision Log as the DCN form will be used for this. Corrected an error in the Socket Control Header description		DCN #E22203, 11 Feb 2003
B	DST/JEE/DAO 8/5/2003	Modified elements of the connection protocol Modified ARR error code definitions Moved all requirements from 1.4.3 and 1.4.4 to the respective command and telemetry sections Modified SSH configuration / operation details Updated many structure diagrams Substantial re-organization of the Telemetry Section but no interface changes Support Products section significantly updated		

ITAR 125.4(b)(3) Exemption of General Applicability applies IAW NASA Contract NAS5-97271, Task Order 28j, dated 5/31/02

August 5, 2003

THE JOHNS HOPKINS UNIVERSITY

APPLIED PHYSICS LABORATORY

11100 JOHNS HOPKINS ROAD, LAUREL, MARYLAND 20723-6099

STEREO MISSION OPERATIONS CENTER (MOC) to PAYLOAD OPERATIONS CENTER (POC) and to STEREO SCIENCE CENTER (SSC) INTERFACE CONTROL DOCUMENT (ICD) --DRAFT--

FSCM NO.	SIZE	DRAWING NO.	REV.
88898	A	7381-9045	B
SCALE	DO NOT SCALE PRINT	SHEET 1 of 68	

INSERT SIGNATURES HERE

TABLE of CONTENTS

TABLE of FIGURES

TABLE of TABLES

1 INTRODUCTION

1.1 PURPOSE

This document defines the command, telemetry, and support product interfaces between the Mission Operations Center (MOC) at The Johns Hopkins University/Applied Physics Laboratory (JHU/APL), the Payload Operations Centers (POCs), and the STEREO Science Center (SSC).

1.2 BACKGROUND

The POCs for the four STEREO investigations (SECCHI, IMPACT, SWAVES, and PLASTIC) consist of the necessary electronics, equipment, software, and computer-based systems to facilitate complete instrument checkout and to provide integration support during the final integration and test (I&T) phase with the two STEREO Observatories. In addition, some elements of the POCs will continue to provide operational support post-launch. The elements of the POCs that continue are those that provide instrument commands to, and accept telemetry from, the STEREO MOC. The interfaces described within this document apply to the instrument checkout I&T, Environmental Testing, and post launch operations phases of the mission. The differences for these time periods will be clearly described within this document.

The POCs' instrument operations will be decoupled from the spacecraft operations for the STEREO mission. The goal of this decoupled operations concept is to reduce system complexity and cost. The Spacecraft and POCs' instruments will be operated independently of each other with only a few exceptions.

The interfaces between the POCs and the MOC include those supporting commands sent to the STEREO spacecraft as well as those supporting telemetry data received from the STEREO spacecraft. The POCs have the option of interfacing with the STEREO spacecraft via two separate interfaces: via the APL-MOC command/telemetry interface or optionally via direct access to an instrument test port during I&T. An investigation's POC supports access to these two interfaces via two functions which are defined as the *direct access* function and the *command/telemetry* function.

During I&T the *direct access* interface function exists to provide stimuli input via a direct test connection using a test port (if the investigation has implemented such) and to feed back stimulus and instrument indicators. Usage of direct access will be further detailed within appropriate integration & test documentation and is not described here. Direct-test connection access *may* also exist for instruments to allow for rapid software uploads and for ground-only configuration (pre-launch calibration or alignment checks) testing. Direct access may be limited during certain phases of I&T and is not available after shipment of the observatories to the launch sites.

The *command/telemetry* interface function forwards instrument commands to the spacecraft and receives telemetry from the spacecraft via the APL MOC. During the early Integration and Test phase, the POCs will interface with the APL MOC for integration testing. During the Mission Simulation phase of I&T, the full functionality of the MOC interface should be available to provide end-to-end checkout of the command & telemetry streams. The *command/telemetry* interface continues to be maintained throughout the life of the STEREO mission. It is required for the post-launch control of instrument operations. The *command/telemetry* interface functions will be further detailed in later sections.

The SSC to MOC interface consists of two different interfaces: the Space Weather data interface and the Long Term archive interface. The Space weather data interface will be available at the same time as the *command/telemetry* interface. The Long Term archive interface will be available in time to provide end-to-end checkout of the SSC to MOC interface. The SSC to MOC interfaces are described in detail in later sections.

1.3 DOCUMENT ORGANIZATION

This document is divided into two main sections. The first section addresses the document's purpose and scope and provides an overview of the interfaces. The second section describes in detail the interfaces between the POCs and the MOC for the transmission of instrument commands to the spacecraft and the receipt of instrument telemetry data from the spacecraft. The second section also describes in detail, the interfaces between the SSC and the MOC for the transfer of real-time space weather data and the transfer of the instrument telemetry data for storage in the long term archive. In addition, the second section of this document includes a summary description of the data products which will be exchanged between the MOC and POCs.

1.4 REFERENCES

1. STEREO Mission Requirements Document, NASA GSFC, 460-RQMT-0001, August 2000
2. STEREO Mission Operations Concept of Operations, JHU/APL 7381-9020
3. STEREO SECCHI Investigation, Interface Control Document, JHU/APL 7381-9011
4. STEREO IMPACT Investigation, Interface Control Document, JHU/APL 7381-9012
5. STEREO SWAVES Investigation, Interface Control Document, JHU/APL 7381-9013
6. STEREO PLASTIC Investigation, Interface Control Document, JHU/APL 7381-9014
7. CCSDS 200.0-G-6: Telecommand Summary of Concept and Service. Green Book. Issue 6. January 1987
8. CCSDS 202.0-B-2: Telecommand Part 2 -- Data Routing Service. Blue Book. Issue 2. November 1992(Reconfirmed June 1998.) This Recommendation has been adopted as ISO 12172:1998.

9. CCSDS 202.1-B-1: Telecommand Part 2.1 -- Command Operation Procedures. Blue Book. Issue 1. October 1991. This Recommendation has been adopted as ISO 12173:1998
10. CCSDS 301.0-B-2: Time Code Formats. Blue Book. Issue 2. April 1990. This Recommendation has been adopted as ISO 11104:1991.
11. CCSDS Global Spacecraft Identification Field Code Assignment Control Procedures. Blue Book. Issue 1. November 1998.
12. STEREO EA and CDH Software Requirements, JHU/APL 7381-9226
13. STEREO Mission Operations Center (MOC) Data Products Document, JHU/APL 7381-9047

1.5 INTERFACE SUMMARY

This summary defines the equipment needed to support the interfaces during the different phases of the STEREO mission. In addition, the summary provides a high level overview of the verbal, command, telemetry, and data product interfaces. Figure 1 depicts a high level context diagram of the interfaces through the STEREO ground systems.

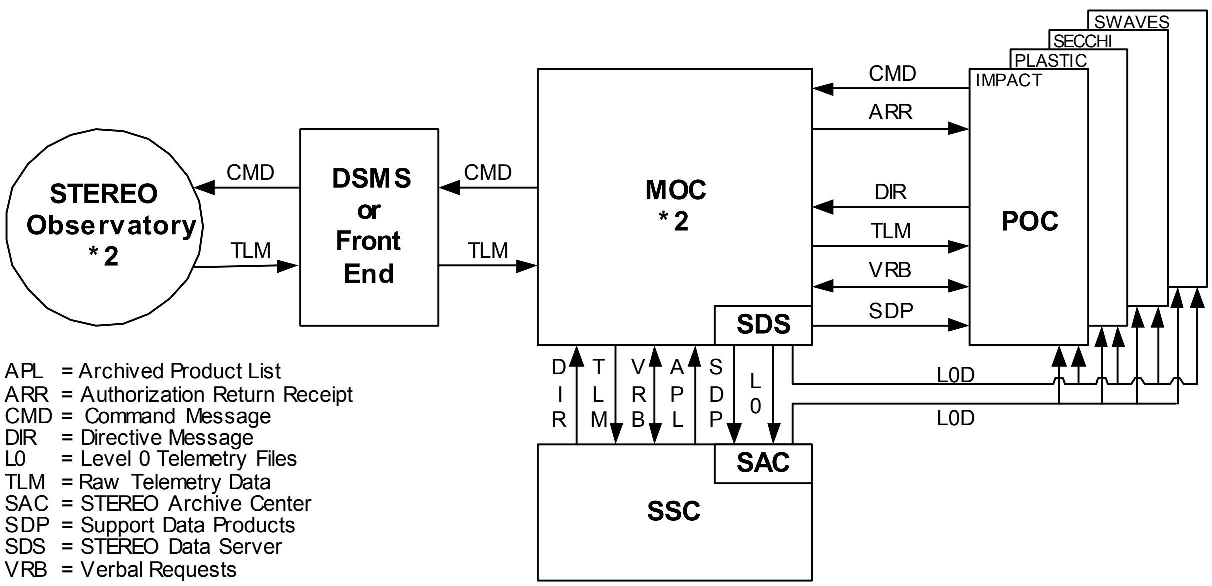


Figure 1. STEREO Command and Telemetry Interface Context Diagram

1.5.1 EQUIPMENT

One set of each POC's equipment will be provided by the associated investigation team to allow I&T of the two STEREO observatories. It is also expected that POC personnel will support mission simulations that are performed during I&T. The SSC will provide access to an interface with the MOC to allow for I&T of the MOC to SSC interface. The location and access to the equipment for these interfaces will be determined prior to the beginning of I&T.

1.5.2 VERBAL INTERFACES

The MOC/POC and MOC/SSC interfaces are normally computer-controlled. However, in order to ensure smooth operations and coordination between the MOC and the POCs and SSC, a verbal interface (VRB) is necessary. This interface will consist of routine meetings, ad-hoc meetings, and communications in person, via telephone, telecon, and/or televideo. The routine meetings will be scheduled daily during I&T through Instrument Commissioning, and then scheduled weekly through the Normal Operations Phase of the mission. Ad-hoc meetings will consist of phone calls and will be used whenever necessary for continued operations. In particular, emergencies on either the spacecraft or instruments will necessitate an immediate phone call or page to facilitate quick resolution of the emergency. Voice communications will be handled such that there will never be a time when operations staff, either spacecraft or instrument, cannot be reached. The verbal interface is described in detail in Section 2.3.

1.5.3 COMMAND INTERFACE OVERVIEW

The IMPACT, PLASTIC, SECCHI, and SWAVES POCs will send Supplemented Command Messages (SCMs) to the MOC which contain the information needed to configure and control their instruments. For each SCM received, the MOC will apply simple verification rules to the header contents and construction, and will send an Authorization Return Receipt (ARR) message back to the POC's facility indicating the message status.

SCMs that pass the verification tests are maintained in queues by the MOC. If a POC wants to delete SCMs held in their queue, the POC verbally sends the MOC a command flush request (CFR).

Based on the command delivery time information included in the SCM header, the MOC will forward the SCMs to the I&T Front End or to the Deep Space Mission System (DSMS) interface for transfer to the spacecraft. Finally, the Command and Data Handling (C&DH) process on the spacecraft will forward the commands to the 1553 bus for instrument retrieval. Actual command execution success or failure will be indicated in instrument telemetry telltales. The command interface between the POCs and the MOC is described in detail in Section 2.4.

1.5.4 TELEMETRY INTERFACE OVERVIEW

The spacecraft will transmit real-time telemetry consisting of spacecraft housekeeping data, space weather packets, and limited science instrument data to the DSMS for forwarding to the MOC as the telemetry is produced. The spacecraft will record housekeeping information, pertinent engineering and attitude data, and science instrument data on the solid-state recorder (SSR) for later transmission to the DSMS and subsequent off-line forwarding to the MOC.

During a scheduled track with the DSMS, the spacecraft will relay to the ground the real-time telemetry and the contents of the SSR (Dump Telemetry). The DSMS will immediately forward downlinked real-time telemetry to the MOC during the track and will save all of the real-time and SSR dump telemetry to files. After the track, the DSMS will transfer the files to the MOC via FTP. The MOC will use the real-time spacecraft telemetry to monitor the spacecraft health and spacecraft command delivery status.

The MOC will provide the POCs access to telemetry being received in real-time from the DSN via a TCP/IP connection. Data delivered via this interface is "best effort"; i.e., it will drop packets if the POC does not accept them as fast as they are arriving at the MOC. The POCs specify which telemetry is desired via a series of directive messages (DIR). The POCs can request spacecraft housekeeping data as well as instrument data via these interfaces. The MOC will make available descriptions of the format of the spacecraft housekeeping packets to aid in interpretation.

The MOC will provide the POCs access to archived instrument telemetry via a TCP/IP connection or FTP file transfer. The POCs and the SSC may establish a TCP/IP connection with the MOC to retrieve the telemetry from the archive. The POCs will send the MOC directive messages, which specify what telemetry will be sent on the socket. The MOC will send the requested archived real-time or dump telemetry to the POCs and to the SSC.

On a daily basis, the MOC will produce files containing cleaned and merged (Level 0 [L0]) telemetry for each instrument. The POCs will retrieve the L0 files from the STEREO Data Server (SDS) via FTP. Prior to the deletion of the instrument real-time and SSR dump telemetry from the archive, the MOC will produce a final version of the L0 files for each POC. This is intended to provide the most complete data set possible, since it would include telemetry for a specific day that could have been downlinked any time in the last 30 days.

The SSC will retrieve the final version of the L0 files via FTP for long term storage in the STEREO Archive Center (SAC). Then, the SSC will generate an Archived Product List (APL), which lists the L0 files that were transferred to the long term archive. The MOC will retrieve the APL via FTP from the SSC, and will verify that the L0 files were successfully stored in the SAC. If the L0 files were archived in the SAC, the MOC will delete the L0 files and the instrument real-time and dump telemetry from the short term archive. If the POCs need an L0 file after deletion from the MOC, the POCs will retrieve the files from the SAC. The telemetry interfaces between the POC, SSC, and the MOC are described in detail in Section 2.5.

1.5.5 SUPPORT DATA PRODUCTS OVERVIEW

The Support Data Products (SDPs) consist of a series of files that are used by MOC and POC personnel for planning, determining status, and supporting the operations of the STEREO spacecraft. These products will be stored on the SDS for retrieval by the POCs and SSC. A list of the available files will be accessible via a web browser, and the files will be transferred between the systems via FTP. A list of the SDPs is included in Section 2.6. The SDPs will be described in detail in the STEREO MOC Data Products Document (Reference 13).

2 INTERFACES

2.1 INTERFACE DATA FLOW DIAGRAMS

A high level data flow diagram for the interfaces between the STEREO MOC and the POCs is shown in Figure 2. The verbal interface between the MOC and the POCs is described in Section 2.3. The command interface between the MOC and the POCs includes the transmission of the SCM and ARR messages, which are described in Section 2.4. The telemetry interface includes the transmission of the Telemetry Directives (DIR), Supplemented Telemetry Packets (STP), Payload Telemetry Packets (PTP), and L0 Files, which are described in Section 2.5. The SDPs exchanged between the MOC and the POCs are described in Section 2.6.

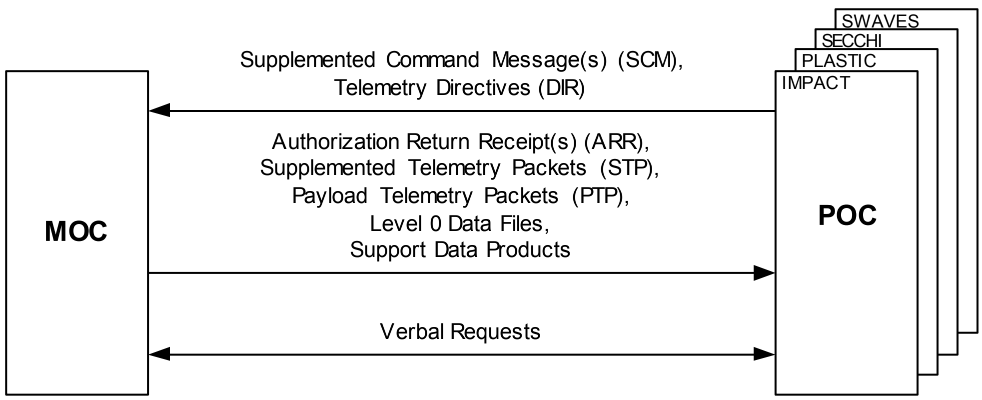


Figure 2. MOC to POCs Data Flow Diagram

A high level data flow diagram for the interfaces between the STEREO MOC and the SSC is shown in Figure 3. The verbal interface is described in Section 2.3. The telemetry interface includes the transmission of the DIRs, APL, STPs, PTPs, and L0 Files which are described in Section 2.5.

2.2 INTERFACE REQUIREMENTS

The interfaces described in this document are defined by four characteristics: protocol, format, content, and file naming convention where filenames are needed. The characteristics are unique to each interface and are described in the following sections.

In general, the following applies to all of the interface messages unless otherwise specified. Each field within each message has a defined length, and there are no delimiters between the fields. Therefore, if the data stored in the field is not the length of the field, the specified fill character should be used to ensure the field is the full length. In general, the fill character for ASCII strings is a dash (“-“) or space (“ “) character, and the ASCII numeric fields are filled with leading zeros (e.g, “0023”). Any exceptions to these rules will be clearly defined in the following sections.

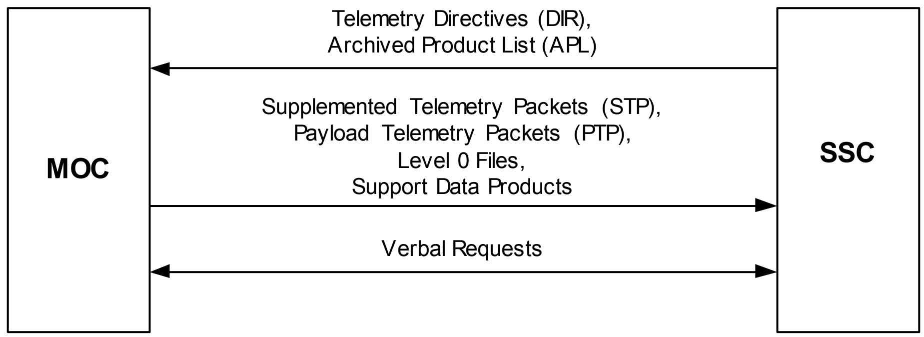


Figure 3. MOC to SSC Data Flow Diagram

2.3 VERBAL INTERFACE

The POC/MOC and SSC/MOC interfaces are normally computer controlled. However, there shall also exist a verbal interface among the POCs, the SSC, and the MOC personnel. The verbal communication between the MOC and the POCs will occur both routinely and sporadically. Routine meetings shall be scheduled and sporadic meetings shall be convened based on the state of the observatory and as needed. These meetings shall be scheduled such that remote operations personnel can participate via telecon and/or televideo.

Routine meetings shall be scheduled throughout I&T, Early Operations/Instrument Commissioning, and Normal Operations. These meetings shall be scheduled daily from I&T through Early Operations/Instrument Commissioning, and then weekly during Normal Operations. The purpose of the daily meetings shall be to inform the teams of the scheduled activities for the day and to discuss in detail any current problems (schedule or otherwise) with the subsystems or instrument and the plans for resolving the problems. During Normal operations, the weekly meetings shall focus on the long term schedules and resolving any conflicts with spacecraft resources. Of primary concern shall be the scheduling of momentum dumps and instrument calibration maneuvers.

Sporadic meetings shall occur whenever necessary. Call lists shall be provided to all teams and shall include phone numbers and pager numbers. The call list shall allow someone from each team to be reached regardless of the day or time of day. Should an anomaly occur on the spacecraft or an instrument, these lists shall be used to inform affected teams immediately such that resolution can begin without delay.

The POCs may verbally request a command Queue Flush. This Queue Flush Request must be made verbally and once requested, shall result in the deletion of all commands that were in the priority, standard-staging, and/or standard-execution queues for that instrument-observatory combination. The three types of queues and their operation are described later in this document.

2.4 COMMAND INTERFACE

A high level diagram of the Command interface between the MOC and the POCs is depicted in Figure 4 and is described in the following sections.

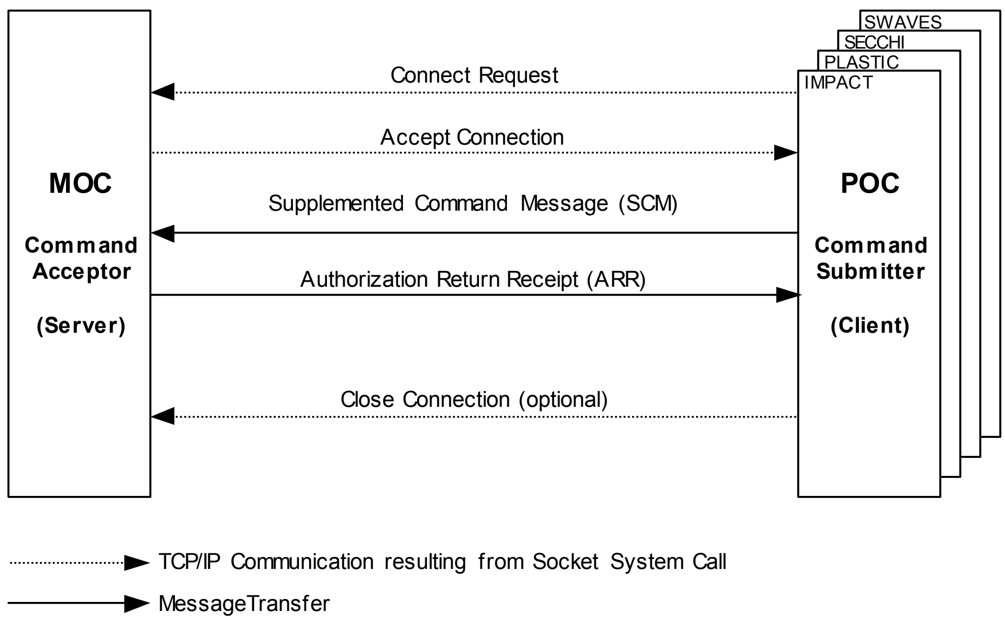


Figure 4. MOC to POCs Command Interface

2.4.1 COMMAND INTERFACE OVERVIEW

During any time periods when the POCs are located remotely from the APL MOC, communications shall require the use of a secure communication support program. The STEREO program has selected the SSH-2 protocol to add security features such as traffic encryption, server authentication, and client authentication. Web sites with detailed information about SSH can be found at http://www.ssh.com and http://www.openssh.org. The SSH server daemon will operate on the APL MOC Command Acceptor machine. The POCs will use SSH client software to initiate an SSH tunnel connection with port forwarding enabled. The tunnel is normally expected to remain established continuously. The SSH operational configuration is described in Section 2.4.7.

Following establishment of the tunnel (if applicable), the POC establishes a TCP/IP connection with the MOC on a port reserved for that POC. The POC can, but does not have to, maintain this connection indefinitely. The MOC will not terminate an established connection unless it detects a connection failure or the POC makes a verbal request to be disconnected.

The SCMs sent from the POCs to control their instruments will be sent via the TCP/IP connection to the MOC. A *transaction* is defined as the transmission by the POC of a single SCM to the MOC and the transmission by the MOC of the corresponding ARR message to the POC. Transactions are atomic; i.e., the POC cannot begin transmission of an SCM prior to receiving the ARR for the previous SCM.

The SCM sent by the POCs will contain a Socket Control Header, Signature Header (name originated from the now-obsolete use of a digital signature), MOC Control Header, and one or more CCSDS telecommand packets. Each CCSDS telecommand packet may contain one or more instrument commands. The MOC does not analyze the data contents of the POCs' telecommand packets.

The POCs will originate all instrument commands to their respective instruments with the exception of all relay commands; e.g., power and deployment actuators. All instrument relay commands will only be originated by the MOC and only transmitted with the prior approval of the respective POC.

During normal operations, the SCMs will be sent to the MOC at least 8 hours before the scheduled DSMS track. During STEREO spacecraft or instrument emergencies, instrument commissioning, and I&T, the MOC will allow the POCs to send SCMs while the MOC is in communication with the spacecraft (either through the Front End or through the DSMS).

The MOC maintains three classes of queues: priority, standard-staging, and standard-execution. There is an instance of each class of queue for each POC. The priority queue holds real-time SCMs (i.e., those with zeros for the delivery enable time). The standard-staging queue holds SCMs with delivery enable times that have not yet been reached, and the standard-execution queue holds SCMs with delivery enable times that have been reached. Any or all of these queues can be flushed as a result of a POC verbal request.

Queues can be enabled and disabled by MOC command. A disabled queue will not allow any commands to be forwarded to the spacecraft. An enabled queue will allow commands to be forwarded to the spacecraft, subject to certain conditions (e.g., the command delivery enable time).

Assuming all queues are enabled, the following servicing approach is used:

1. Initially beginning with IMPACT and proceeding with PLASTIC, SWAVES, and SECCHI, if there is a priority packet waiting to be sent, send it and proceed to the next instrument; if no packet is waiting, proceed to next instrument; when four consecutive queues have been examined with the result that none had packets waiting, switch to servicing the standard-execution queue set. For example, if SWAVES had a packet waiting (which was then forwarded), and SECCHI, IMPACT,

PLASTIC, and SWAVES queues subsequently turned up no waiting packets, note the last queue examined and switch to servicing the standard-execution queue.

2. Initially beginning with IMPACT and proceeding with PLASTIC, SWAVES, and SECCHI, if there is a standard packet waiting to be sent, send it, move the queue pointer to the next instrument queue in the sequence, and return to scan the priority queue set. Otherwise, move the queue pointer to the next instrument queue in the sequence and return to scan the priority queue set.

In addition, the following points are noted:

1. Two of the three classes of queues (priority and standard-execution) are serviced in a pure round-robin fashion. Although the round-robin begins with IMPACT the very first time after the process is started, subsequent servicing is handling on a continuously rotating basis. Because servicing is handled on a packet by packet basis, with interleaving between instruments, no one instrument can monopolize the bandwidth.
2. A full scan of the priority queues is always conducted after each examination of a standard-execution queue, regardless of whether a packet was forwarded from the standard-execution queue.
3. The POC can verbally request that any or all of their queues be flushed (including the standard-staging). In such a case, the files containing the SCMs are deleted.
4. The queuing process maintains knowledge of the enable times of SCMs in the standard-staging queue and will move those to the standard-execution queue no earlier than the enable time and within 5 seconds after the enable time.
5. The queuing process maintains knowledge of the expiration times of SCMs in the standard-staging queue and will remove those from the queue no earlier than the expiration time and within 5 seconds after the expiration time. In no event will the queuing process advance an SCM to the standard-execution queue after the expiration time.
6. The queuing process maintains knowledge of the expiration times of SCMs in the standard-execution queue. The behavior is the same as with the standard-staging queue EXCEPT that once at least one packet from an SCM has been forwarded to the spacecraft, the remainder will be forwarded. That is, the granularity of control is at the SCM level and the queuing process favors completing in-progress SCMs.
7. If transmission of packets from an SCM has begun for a given queue, and the operator disables the queue, the remaining packets will be forwarded. That is, the level of granularity in the disable queue command is the SCM, and the queuing process favors completing in-progress SCMs.
8. If spacecraft packets are ready to be radiated, they are interleaved with instrument packets.

9. All packets within an SCM are sent in the order they appear in the SCM. SCMs with the same Delivery Enable Times are sent based on the SCM message counter value.

2.4.2 COMMAND INTERFACE PROTOCOL

Each Observatory MOC shall establish one unique Command Acceptor listening port for each STEREO POC. During normal operations, there shall be two instances of the Observatory MOC, MOC-A and MOC-B; i.e., one for each STEREO observatory. Therefore each POC shall be responsible for the operation of their instrument on Observatory A through the MOC-A assigned host/port interface, and for the operation of their instrument on Observatory B through the MOC-B assigned host/port interface.

When the MOC instance accepts a connection from a particular POC, it will close the listening port for that POC. Should another connection be attempted by the POC while a connection exists, the second connection attempt will fail. Such a failure will have no effect on the established connection.

A POC may keep a connection active indefinitely. If the POC closes the connection, or the MOC detects a problem with the connection while attempting to send an ARR, the MOC will close the connection and will re-establish a listening port for the POC.

In the event that the Command Acceptor Process fails, the MOC shall restart the process. If the host upon which the process runs fails, the MOC shall start a copy of the process on a backup host. This will require the POC to establish an SSH tunnel with the backup workstation. In the event that Internet communications are unavailable or impractical due to excessive latency, the MOC shall provide for dial-in modem access to the MOC for command delivery. Use of SSH remains required, even with modem use. Modems use PPP to transmit TCP/IP traffic over phone lines. The MOC shall be capable of simultaneously supporting one or more POCs communicating via modem while one or more POCs are communicating via the Internet.

The basic command delivery protocol from the POC's perspective is:

1. POC sends an SCM to the MOC via the established TCP/IP connection.
2. POC reads an ARR from the MOC via the established TCP/IP connection.
3. Repeat

The basic command delivery protocol from the MOC's perspective is:

1. MOC reads an SCM from the POC via the established TCP/IP connection.
2. MOC verifies the values of selected header fields and the result of the attempt to save the SCM locally to a file.
3. MOC sends an ARR to the POC via the established TCP/IP connection with flags set to reflect success or failure, as appropriate.

4. Repeat

The content of the SCM is described in Section 2.4.5.1.

2.4.2.1 TIMING

The command delivery protocol includes timing requirements to support certain functional requirements. The following paragraphs detail the timing requirements and the functional requirements they support.

The following describes the mechanism the MOC uses for determining that the POC sent too many or too few bytes in the SCM:

1. To detect too few bytes, the MOC shall implement time checks following the receipt of the first byte(s) of an SCM (i.e., the data returned on a non-blocking read following a return from a "select" statement indicating socket readability). Following receipt of data, the MOC will wait a configurable period of time (nominally 2 minutes) for the next successful receipt of data, resetting the timer with each successive data receipt, until the expected amount of data have been received. If a timeout occurs prior to the MOC receiving the expected amount of data (as specified in the SCM header), the MOC shall conclude that the POC sent what it thought was the complete data set, and will report via a negative ARR that insufficient data were received.
2. If after reading the expected number of bytes, the MOC detects that additional data remain available for reading, it shall conclude that the POC sent too many bytes, and shall report via a negative ARR that too much data were received. In this case, the MOC will attempt to clear available data from the local TCP buffer prior to sending the ARR, either through an additional "read" action or through a system mechanism to flush the buffer. If additional data had been sent by the POC application and were either buffered at the POC or in transit to the MOC at the time the MOC sends the negative ARR, it is possible that these late arriving data will appear to the MOC as the beginning of a new SCM. Since the MOC determines the expected number of bytes in the SCM from a field in the header, its first action is to try to read the entire header. If less than the header number of bytes appear within the timeout period, the MOC will respond to this case by sending a negative ARR indicating too little data. If at least the header number of bytes appear, the MOC will look in the length field, which will contain some unpredictable value, and will attempt to read that many bytes. This out-of-synchronization condition could result in a conclusion that too many, too few, or just the right number of bytes were received. Should it appear to the MOC that the right number of bytes were received, the basic SCM verification is still likely to fail given the expected random nature of the error. If the POC suspects that an unrecoverable out-of-synchronization situation exists, they should disconnect from the MOC and reconnect.

3. The nominal timeout period (2 minutes) is based on a worst-case estimate of the latency over the Internet. It is not intended to imply support for incremental delivery of SCM bytes from the POC.

4. The POCs shall wait a configurable time period (nominally 5 minutes) from the time of SCM transmission for receipt of a corresponding ARR. This allows for a worst-case 2 minute one-way transit time and worst-case processing delay at the MOC. If no corresponding (i.e., not heartbeat) ARR is received in this time, the POC shall conclude there is a connection problem, disconnect, and attempt to reconnect and resend the SCM. There is no scenario where this type of error could be remedied by simply retransmitting the SCM.

2.4.2.2 MOC PROCESSING

2.4.2.2.1 SOCKET CONTROL HEADER VALIDATION

The MOC shall begin by reading the Socket Control Header, which includes the message type, the message length, and the message counter. The MOC shall verify that the values for the fields are valid ACSII values, the message type is SCM, and the advertised length is within the acceptable range. If the header fails any of these tests, the MOC shall send an ARR with error code “001” to the POC and shall disconnect the POC. The rationale for disconnecting the POC in this case is that it is not possible to guarantee correct interpretation of subsequent data if the header is uninterpretable.

2.4.2.2.2 COMMAND MESSAGE VALIDATION

The MOC shall read from the socket the rest of the command message based on the length of the message specified in the Socket Control Header. If the length of data specified in the socket header is not equal to the total amount of data available on the socket, the error shall be reported to the POCs in the ARR message, and the command message shall not be queued for transmission to the STEREO spacecraft.

The MOC shall perform limited value checking of the SCM. First, the MOC shall perform the following checks of the MOC Control Header fields:

- Facility ID (IMPACT, PLASTC, SECCHI, SWAVES) valid for the port number.
- Spacecraft ID valid for the port number.
- Command Delivery Enable time occurs before the time-out time.
- Command Delivery Time-out time is in the future.
- End of Header indicator found in the MOC Control Header.

Then, the MOC shall check the following fields within each CCSDS telecommand packet:

- There must be at least 6 bytes representing the primary header and at least 1 byte of data
- APID (See Appendix B) in each telecommand packet valid for the Facility ID.
- Total size of the telecommand packets consistent with message length.

The MOC will not examine or make use of the CCSDS telecommand packet sequence count. If the MOC Control Header or telecommand packet fields are invalid, the error shall be reported to the POCs in the ARR message, and the SCM shall not be queued for transmission to the STEREO spacecraft.

2.4.2.2.3 SCM STORAGE

Next, the MOC will save the valid SCMs to disk within the MOC command staging area. The filenames for the MOC SCMs are uniquely identified based on the Facility ID, SCID, enable time, time-out time, and message counter sent in the original SCM. The MOC shall save the most recently received version of the SCM (version is determined by the Facility ID, SCID, enable time, time-out time, and message counter). Therefore, the POC may resend a version of an SCM multiple times, but the commands in the SCM will only be sent to the spacecraft once. If the MOC is unable to save the SCM to disk, an ARR with an appropriate error code shall be generated and sent to the POC.

2.4.2.2.4 ARR GENERATION

The MOC shall create the ARR message, which shall include the Socket Control Header, the ARR Header, which includes the command verification status, and the CCSDS Telecommand Primary Header, as copied from the first Telecommand Packet in the command message. A complete description of the ARR can be found in Section 2.4.5.2. For cases where insufficient SCM data were received to fully populate the ASCII text fields of the ARR Header, this fields, the MOC shall pad the fields with ASCII blanks. For cases where insufficient data were received to fully populate the CCSDS Telecommand Packet Primary Header section of the ARR, the MOC shall pad the remaining space with binary zeros.

The MOC shall write the ARR message to the socket. The POC, which has been waiting for the ARR message, shall read the message from the socket. If a failure occurs when sending the ARR message to the POCs or if verification of the SCM failed, the MOC shall not queue the SCM for transmission to the spacecraft. The MOC shall log the receipt of the SCM, the status of the command validation, the ARR, and the transmission of the SCM to the DSMS or Front End.

If there is a failure in communications on the socket at any time during the session or if the MOC reports an error in the ARR, the POCs shall be responsible for resending the command message.

2.4.2.2.5 COMMAND ORDERING

The MOC shall use the command delivery enable time as the primary determinant for the order the commands are sent to the spacecraft. If the enable time is set to zero, the command shall be designated as a “priority” command. The definition of a priority command in this context, is one that will be forwarded to the observatory at the next available opportunity. These commands may be queued if there is not an existing link to the observatory command system. Note that all priority commands in the queue are radiated before any standard commands.

The MOC shall use the message counter value for SCMs as the secondary determinant for the order the commands are sent to the spacecraft. Since the counter “rolls over” to “1” after “9999”, the MOC shall employ a windowing algorithm to determine the radiation order for SCMs. The window size shall be configurable and shall initially be 5000. This number was chosen because it represents half the available number space of the field. If after ordering the SCMs with a given enable time by ascending message count value, the difference between the message counter values of any two SCMs in the sequence is greater than the window size, then that pair is presumed to define the end point and the starting point, respectively, of the desired radiation order. Note that this algorithm will fail if the POC ever has more than 5000 SCMs with the same enable time stored at the MOC. The POCs can work around this limitation by choosing to advance the enable times of SCMs by one second, either on a per SCM basis, or on a per-block-of-SCMs basis where the block is 5000 or less.

Based on the MOC’s approach for ordering transmission of SCMs by message counter value, the POCs need to evaluate their commanding approach and their handling of negative ARRs to ensure that the end result is what is desired. As an example of an approach that could be problematic, consider the following scenario. Suppose a POC were to send a sequence of 10 SCMs. The POC always waits for the ARR associated with an SCM before sending the next SCM, but does not examine the result code of the ARRs until after the sequence has been sent. At that point, the POC notices that SCM #3 received a negative ARR. Following are some sample POC responses and the resulting outcomes. These examples presume that the enable time has not been reached or that the queue has not yet been enabled and consequently all SCMs sent by the POC are resident at the MOC.

1. The POC corrects the SCM and resends it, but the message counter has continued to advance and the value “11” is used for what should be the third SCM in the sequence. Thus the effective order delivered to the spacecraft would be 1, 2, 4, 5, 6, 7, 8, 9, 10, 3 which is an incorrect result.
2. The POC corrects the SCM and resends the entire sequence, allowing its program to continue to increment the message counter value. Consequently, the MOC will conclude that it is receiving new SCMs and not replacements for earlier ones. The MOC will transmit 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, which is an incorrect result.

3. The POC corrects the SCM and resends it alone, explicitly telling its program to use “3” for the message counter. The effective order delivered to the spacecraft would be 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 as desired.
4. The POC corrects the SCM, resends the entire sequence, but instructs its program to reset the initial message counter to “1”. The MOC receives the SCMs and replaces the existing ones since they match on enable time, expiration time, and message counter (with the exception of the third SCM which is simply accepted). The effective order delivered to the spacecraft would be 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 as desired.

2.4.2.2.6 COMMAND TRANSMISSION

Based on the command delivery enable time and the time-out information provided in the SCM MOC Control Header, command packet(s) from the SCM shall be sent via the Front End during I&T and via the DSMS following launch for delivery to the instrument. The definitions for the fields in the time block of the MOC Control Header can be found in Table 1. The rules governing the usage of the command delivery enable time, command delivery time-out, and command delivery delta time-out are listed in Table 2.

2.4.2.2.7 COMMAND RECEIPT VERIFICATION

Indications that the commands were received by the spacecraft may be gleaned from telemetry received from the spacecraft, and include the instrument stored command buffer report and the command history report. The Instrument Stored Command Buffer report (See Section 2.6.24) indicates which command packets are stored on the spacecraft and are queued for execution. The Telecommand Packet History report (See Section 2.6.25) indicates whether the packet was successfully received by the C&DH.

Table 1. Command Delivery Time Definitions

Time Field	Time Definition
Enable Time	Indicates to the MOC the time that the MOC can start trying to send the POC’s command to the spacecraft.
Time-Out Time	Indicates to the MOC the time that the MOC should stop trying to send the POC’s command to the spacecraft.
Delta Time-Out	Duration in seconds that will be added to the command delivery enable time to compute the time that the MOC should stop trying to send the POC’s command to the spacecraft. Supersedes the command delivery time-out.

Table 2. Command Delivery Rules

Enable Time	Time-out Time	Delta Time-out	Rule
Zero	Dashes	Dashes	Enable time is MOC Receipt Time; time-out is infinite; ONLY Used for priority commanding.
Zero	Dashes	Present	Enable time is MOC receipt time; time-out is receipt time plus Delta Time-out; ONLY Used for priority commanding.
Zero	Present	Dashes	Enable time is MOC receipt time; time-out is Time-out Time; ONLY Used for priority commanding.
Zero	Present	Present	Enable time is MOC receipt time; time-out is receipt time plus Delta Time-out; ONLY Used for priority commanding.
Present	Dashes	Dashes	Enable time is Enable Time; time-out is infinite.
Present	Dashes	Present	Enable time is Enable Time; time-out is enable time plus Delta Time-out.
Present	Present	Dashes	Enable time is Enable Time; time out is Time-out Time.
Present	Present	Present	Enable time is Enable Time; time-out is enable time plus Delta Time-out.

*Zeros (13 ASCII characters "0000000000000") for the enable time (time sent to spacecraft) indicates it is considered a priority command and will be queued for the first available uplink. Dashes for the time-out and Delta time-out (13 ASCII characters "-------------" or 5 ASCII characters "-----") indicate the field is not used.

2.4.2.2.8 C&DH COMMAND HANDLING

Note: C&DH requirements are captured in the EA and CDH Software Requirements Document (Reference 12). A summary of the behavior with respect to instrument commands is given below for the convenience of the reader, but these do not represent formal C&DH requirements.

Onboard the Spacecraft the C&DH processor will receive command packets destined for the instruments and either forward them immediately or store the command packets in the on-board Instrument Stored Command Buffer. (NOTE: The SECCHI POC commands will be immediately forwarded and will not be stored in the Instrument Stored Command Buffer.) The C&DH will forward these command packets to the instrument DPU via the 1553 bus based on the CCSDS telecommand packet secondary header command routing time and the primary header sequence count. If there is no secondary header within the telecommand packet or if the routing time is equal to zero, the command packet will be immediately forwarded to the instrument via the 1553 Bus. If the routing time is in the past, the C&DH will reject the command. For commands stored in the Instrument Stored Command Buffer, if the command routing time is the same for multiple command packets, the packets are sent to the instrument DPU in the order of the sequence count.

2.4.3 COMMAND FLUSH REQUESTS

A POC operator may verbally request the MOC to manually flush commands from the MOC command queue or from the spacecraft on-board Instrument Stored Command Buffer. The POC operator will tell the MOC Spacecraft Specialist when the MOC command queue and/or Instrument Stored Command Buffer should be flushed.

Upon a request to flush the MOC command queue, the MOC will delete all files which contain the instrument's facility identifier (SECCHI, SWAVES, IMPACT, or PLASTIC) from the specified MOC command queue (i.e., priority, standard-staging, or standard-execution) for the observatory.

Upon a request to flush the commands from the Instrument Stored Command Buffer, the MOC will send the appropriate command to the C&DH on the spacecraft either via the Front End or DSMS. The C&DH will execute the flush request immediately, and will delete from the Instrument Stored Command Buffer all of the commands for the specified instrument based on the subsystem identifier (4 = IMPACT, 6 = PLASTIC, and 10 = SWAVES) contained within the flush request. NOTE: The SECCHI POC will not use the Instrument Stored Command Buffer, and therefore will not request a flush of the buffer.

2.4.4 COMMAND CAPACITY

During normal operations, the POCs and MOC can send a combined maximum of 1.8 Megabits (125 bps * 4 hours/day) of instrument and spacecraft commands per day per observatory without MOC intervention. Due to the relatively low command uplink rate, the POCs should minimize their daily command volume. In some instances upload of instrument or observatory software might be needed, and 7.2 Megabits (500 bps * 4 hours/day) of commands per day per observatory shall be allowed. These situations shall be handled on a case-by-case basis between the MOC and the POCs teams via phone calls, schedule plans, and weekly planning meetings. The POCs will be informed at these meetings if a heavy spacecraft load is expected, and the Track Plans (See Section 2.6.28) will provide a plan of the events for all tracks within a week. Software loads at the higher data rate should ideally be coordinated at least two weeks in advance.

2.4.5 COMMAND MESSAGE FORMATS

2.4.5.1 SUPPLEMENTED COMMAND MESSAGES

The SCM formats are shown in Figure 5. The format of the CCSDS telecommand packet, which is part of the command message, is shown in Figure 6. The attributes, including the data size, type, and values for each field within the command messages and the telecommand packet are defined in Table 3 and Table 4, respectively.

The maximum CCSDS telecommand packet length allowed for STEREO shall be 1088 bytes, with a 6 byte primary header, a 6 byte secondary header, and 1076 bytes for the command data. Each SCM shall contain at least 1 and no more than 99 CCSDS telecommand packets.

SUPPLEMENTED COMMAND MESSAGE (SCM)

Socket Control Header (SCH)			Signature Header		MOC Control Header												CCSDS Telecmd Packets
								Cmd Delivery Time Block			Command Description Block				End of Header		
Message Type	Message Length (excludes SCH length)	Message Counter	Command Message Format Version	Spare	Facility ID	SC ID	# Packets	Enable Time	Time-out Time	Delta Time-out	Script Line #	Script Version #	Script Name	Cmd Descript			
"SCM-"	"000126" to "107831"	"0001" to "9999"	"00"	Spare "000"	"SWAVES" "IMPACT" "PLASTC" "SECCHI"	"EA" ahead "EB" behind	"01" to "99"	YYYYDOYHH MMSS or "0000000000000"	YYYYDOYHH MMSS or "------------"	"00000" to "99999" or "-----"	"0000" to "9999"	"0000" to "9999"	e.g., "************************turn-on"	e.g., " Step_V_5mV"	^L (0x0C)		1 to 99 packets
32 bits	48 bits	32 bits	16 bits	24 bits	48 bits	16 bits	16 bits	104 bits	104 bits	40 bits	32 bits	32 bits	256 bits	256 bits	8 bits		Variable 63-8,704 bits per packet
112 bits, **14 bytes**			40 bits, **5 bytes**		912 bits, **114 bytes**												
															min		7 bytes
															max		107,712 bytes

Figure 5. Supplemented Command Message (SCM)

Primary Header								Secondary Header			App Data
Packet Identification					Sequence Control		Packet Length	Cmd Routing Time		Spare	
Version Number	Type	Secondary Header Flag	Application Process ID: Subsystem Identifier	Application Process ID: Data Format ID	Sequence Flags	Sequence Count		Seconds since Epoch	Sub-seconds (1/256)		
000	1	1 = Hdr 0 = No hdr	0 C&DH 2 G&C 4 IMPACT 6 PLASTIC 8 SECCHI 10 SWAVES	See table	11 = standalone 01 = first 00 =continue 10 = last	decimal range = 0 to 16, 383	# Bytes in Sec Hdr + App Data - 1	epoch is Jan 1, 1958 00:00:00 UTC	4 msec count	00000000	User specified
3 bits	1 bits	1 bits	4 bits	7 bits	2 bits	14 bits	16 bits	32 bits	8 bits	8 bits	Variable 8-8,608 bits
48 bits, 6 bytes								48 bits, 6 bytes			1076 bytes max
1088 bytes max											

Figure 6. CCSDS Telecommand Packet Data Structure

Table 3. Supplemented Command Message Format

SUPPLEMENTED COMMAND MESSAGE				
Contents	Size (bits)	Size (bytes)	Type	Units/Range
SOCKET CONTROL HEADER				
Message Type	32	4	ASCII	"SCM-" fill character is a dash "-"
Message Length	48	6	ASCII	Decimal Length (in bytes) of Socket Message, excluding the socket control header length; valid range: "000126" to "107831"
Message Counter	32	4	ASCII	Decimal Counter used to track command message; valid range: "0001" to "9999"; "0000" is invalid.
SIGNATURE HEADER				
CMD Message Format Version	16	2	ASCII	"00" = Initial Version
SPARE	24	3	ASCII	"000"
MOC CONTROL HEADER				
Facility ID	48	6	ASCII	"SWAVES", "IMPACT", "PLASTC", "SECCHI"
Spacecraft ID	16	2	ASCII	Ahead = "EA", Behind = "EB"
Decimal Number of Packets	16	2	ASCII	valid range: "01" to "99"
COMMAND DELIVERY TIME BLOCK				
Enable Time	104	13	ASCII	GMT Time (YYYYDOYHHMMSS) when MOC can start sending cmd to spacecraft, all zeros if real-time; Examples: "0000000000000" "2002171235000" (June 20, 2002 11:50 pm 00 seconds)
Time-out	104	13	ASCII	GMT Time (YYYYDOYHHMMSS) when MOC stops sending cmd to spacecraft, all dashes if not used; Examples: "-------------" "2002171235959"
Delta Time-out	40	5	ASCII	Decimal Number seconds after enable time when MOC stops sending cmd to spacecraft, all dashes if not used; valid range: "00000" to "99999"; Examples: "-----", "86400"
COMMAND DESCRIPTION BLOCK				
POC Script Line	32	4	ASCII	valid range: "0000" to "9999"
POC Script Version	32	4	ASCII	valid range: "0000" to "9999"
POC Script Name	256	32	ASCII	Name of the script, fill character is a space " ". Example:" turn_on"
POC Command Description	256	32	ASCII	Description of the cmd, fill character is a space " ". Example: " Step_V_5mV"
End of MOC Header Indicator	8	1	ASCII	"^L" (Control L) (0x0C is the Hex value of the binary equivalent)
CCSDS Telecmd Packets	Variable	Variable Min = 7 bytes	Binary	1 to 99 Command Packets See CCSDS Telecommand Packet Table
TOTAL SIZE Minimum	**1120**	**140**		

Table 4. CCSDS Telecommand Packet Format

CCSDS TELECOMMAND PACKET				
Contents	Size (bits)	Size (bytes)	Type	Units/Range/Value
PRIMARY HEADER				
PACKET IDENTIFICATION				
Version Number	3	0.375	Binary	000
Type	1	0.125	Binary	"1" designates a Telecommand Packet
Secondary Header Flag	1	0.125	Binary	0 = No Secondary Header 1 = Secondary Header
APPLICATION PROCESS ID				
Subsystem Identifier	4	0.500	Binary	0 = 0000 = C&DH 2 = 0010 = G&C 4 = 0100 = IMPACT 6 = 0110 = PLASTIC 8 = 1000 = SECCHI 10 = 1010 = SWAVES
Data Format ID	7	0.875	Binary	See APID assignment table
SEQUENCE CONTROL				
Sequence Flags	2	0.250	Binary	11 = Standalone Packet
Sequence Count	14	1.750	Binary	Decimal range = 0 to 16,383 To ensure delivery order, increment this counter
Packet Length	16	2.000	Binary	Number of bytes in Secondary Header Fields + Application Data Field **minus 1**
SECONDARY HEADER (Optional) (See NOTE 1)				
COMMAND ROUTING TIME (See NOTE 2)				
Seconds since Epoch	32	4.000	Binary	Number of seconds since Epoch (Jan 1, 1958 0:00:00 UTC), All zeros indicates forward immediately
Subseconds	8	1.000	Binary	Subseconds (1/256 second)
Spare	8	1.000	Binary	00000000
APPLICATION DATA				
Application Data (1 - 1076 bytes)	variable	variable	Binary	No restrictions
TOTAL SIZE Minimum	**104**	**13**		
TOTAL SIZE Maximum	**8663**	**1083**		

NOTE 1: If there is no secondary header or the secondary header fields are set to all zeros, the command is immediately forwarded to the instrument DPU by the observatory C&DH

NOTE 2: The Command Routing Time is the time the C&DH forwards the command to the instrument DPU.

NOTE 3: A STEREO Command Packet (CP) is a specialization of the Telecommand Packet defined in CCSDS 203.0-B-1, ¶5.2.

NOTE 4: Any included Secondary Header is non-CCSDS defined; See CCSDS 203.0-B-1, ¶5.2.2

NOTE 5: The maximum length here is limited by the STEREO program. A higher limit of 65,532 bytes is allowed by CCSDS 203.0-B-1, ¶5.2.1.3.

NOTE 6: The Application Process ID subdivision into Subsystem Identifier Fields & Data Format ID is STEREO specific

2.4.5.2 AUTHORIZATION RETURN RECEIPT MESSAGE

The ARR message structure is shown in Figure 7. The attributes, including the data size, type, and values, for each field in the ARR message are defined in Table 5. The ARR verification flag values, which indicate the status of the authentication of the command message, are listed in Table 6.

AUTHORIZATION RETURN RECEIPT MESSAGE

Socket Control Header			Authorization Message											
Message Type	Message Length	Message Counter	ARR Header				CCSDS Telecmd Packet Primary Header Copied from Command Message							
			Accept or Failure Time	Verify Flag	Command Descript	End of Header	Packet Identification					Sequence Control		Packet Length
							Version Number	Type	Secondary Header Flag	Application Process ID		Sequence Flags	Sequence Count	
										Subsystem Identifier	Data Format ID			
"ARR-"	"000055"	Copied from Command Message	YYYYDOY HHMMSS	See Verify Flag Table	Copied from Command Message	^L (0x0C)	Copied from Command Message	Copied from Command Message	Copied from Command Message	Copied from Command Message	Copied from Command Message	Copied from Command Message	Copied from Command Message	Copied from Command Message
32 bits	48 bits	32 bits	104 bits	24 bits	256 bits	8 bits	3 bits	1 bits	1 bits	4 bits	7 bits	2 bits	14 bits	16 bits
112 bits			392 bits				48 bits							
552 bits / 69 bytes														

Figure 7. Authorization Return Receipt Data Structure Diagram

Table 5. Authorization Return Receipt Format

AUTHORIZATION RETURN RECEIPT MESSAGE				
Contents	Size (bits)	Size (bytes)	Type	Units/Range
SOCKET CONTROL HEADER				
Message Type	32	4	ASCII	"ARR-" fill character is a dash
Message Length	48	6	ASCII	Decimal Length (in bytes) of Socket Message, excluding the socket control header length, "000055"
Message Counter	32	4	ASCII	Decimal Count Copied from the corresponding Command Message
AUTHORIZATION MESSAGE				
ARR HEADER				
Acceptance or Failure Time	104	13	ASCII	GMT time: YYYYDOYHHMMSS
Verify Flag	24	3	ASCII mutually exclusive	"000" to "019" See ARR Verification Flag Table
Command Description	256	32	ASCII	Copied from corresponding Command Message
End of Header Indicator	8	1	ASCII	"^L" (Control L) (0x0C is the Hex value of the binary equivalent)
CCSDS TELECOMMAND PRIMARY HEADER				
PACKET IDENTIFICATION				
Version Number	3	0.375	Binary	Copied from cmd msg (See Note 1)
Type	1	0.125	Binary	Copied from cmd msg (See Note 1)
Secondary Header Flag	1	0.125	Binary	Copied from cmd msg (See Note 1)
APPLICATION PROCESS ID				
Subsystem Identifier	4	0.500	Binary	Copied from cmd msg (See Note 1)
Data Format ID	7	0.875	Binary	Copied from cmd msg (See Note 1)
SEQUENCE CONTROL				
Sequence Flags	2	0.250	Binary	Copied from cmd msg (See Note 1)
Sequence Count	14	1.750	Binary	Copied from cmd msg (See Note 1)
Packet Length	16	2.000	Binary	Copied from cmd msg (See Note 1)
TOTAL SIZE (Bits)	**552**	**69**		

Note 1: These fields are copied from the Primary Header of the "FIRST" CCSDS Telecommand Packet in the corresponding Supplemented Command Message.

Table 6. ARR Verification Flag Values

ARR Verification Flag	
Number	**Description**
"000"	Success
"001"	Invalid Socket Control Header
"002"	Packet size exceeds 1088 bytes
"003"	Number of packets received does not match the number of packets specified in the MOC control header
"004"	Heartbeat (NO OP)
"005"	Command Message Format and Version not supported by the MOC
"006"	The amount of data available on the socket is more than the length of message
"007"	The amount of data available on the socket is less than the length of message
"008"	Invalid Facility ID (Not IMPACT, PLASTC, SECCHI, or SWAVES)
"009"	Invalid Facility ID for the current socket port number
"010"	Invalid Spacecraft ID
"011"	Command delivery enable time after timeout time
"012"	Command delivery timeout time in the past
"013"	End of MOC control header missing
"014"	CCSDS telecommand packet APID invalid for the Facility ID
"015"	Sum of CCSDS telecommand packet lengths inconsistent with length of message received
"016"	Internal MOC processing error (e.g., could not save message to disk, socket read failure)
"017"	A packet is less than the minimum required length
"018"	The packet data received is less than the number of bytes specified by the packet length
"019"	MOC will not accept any more SCMs

2.4.6 EMULATOR, I&T MOC, AND MOC COMMAND INTERFACE DIFFERENCES

Instrument commanding via the emulator will differ from the I&T MOC and the MOC in that the emulator implements a simpler version of the full commanding functionality. The emulator shall process all commands as priority commands and shall immediately forward the commands to the instrument. There shall be no delayed command storage capability and no queuing of commands within the emulator. There shall be minimal checking of the SCM and regardless of the content of the SCM, the emulator shall return an ARR that indicates a successful SCM. The purpose of the ARR when commanding through the emulator is solely to make the user familiar with receiving this receipt and verifying its format and readability. In addition, the emulator shall only accept SCMs and shall not support the use of SSH. If an error occurs during commanding via the emulator and the command is not passed on to the instrument, the user must review the emulator command log to determine the probable cause of the problem.

The I&T MOC operational mode assumes that the POCs operate within the APL firewall and that SSH is not used. However, during Mission Simulations, SSH will be used. The MOC will coordinate with the POCs when SSH is to be used. After the launch of STEREO, SSH shall always be used when POCs are commanding from remote locations.

2.4.7 SSH2 PROTOCOL DETAILS

The APL MOC system administrators shall set-up the firewall rules to the Command Acceptor Workstation such that external access shall be disabled for most services. Access shall be allowed through the SSH port 22. Access shall only be allowed from designated IP addresses. Therefore the POCs shall be required to register client IP addresses with the MOC ahead of time. SSH access shall be restricted to POC user accounts. Public-key authentication with passphrases shall be used for client authentication. User accounts on the MOC server shall use "restricted shell." Access time, remote IP address, and Usernames shall be logged in /var/log/authlog. The MOC server shall not have a .rhosts file. Users shall be required to enter passwords each time they connect for an SSH session.

The POCs shall register their machine type and the version of SSH2 ahead of time with the APL MOC. POCs shall register their host name and IP addresses with the APL MOC. POCs shall receive their user log-in information from the APL MOC.

Only processes on a POC client workstation hosting the tunnel will be able to access the POC-MOC tunnel. Additional security measures shall be taken for each client computer. Usernames and passwords shall be required for access.

The allowed commanding server-listening and client-tunnel ports are shown in Figure 8.

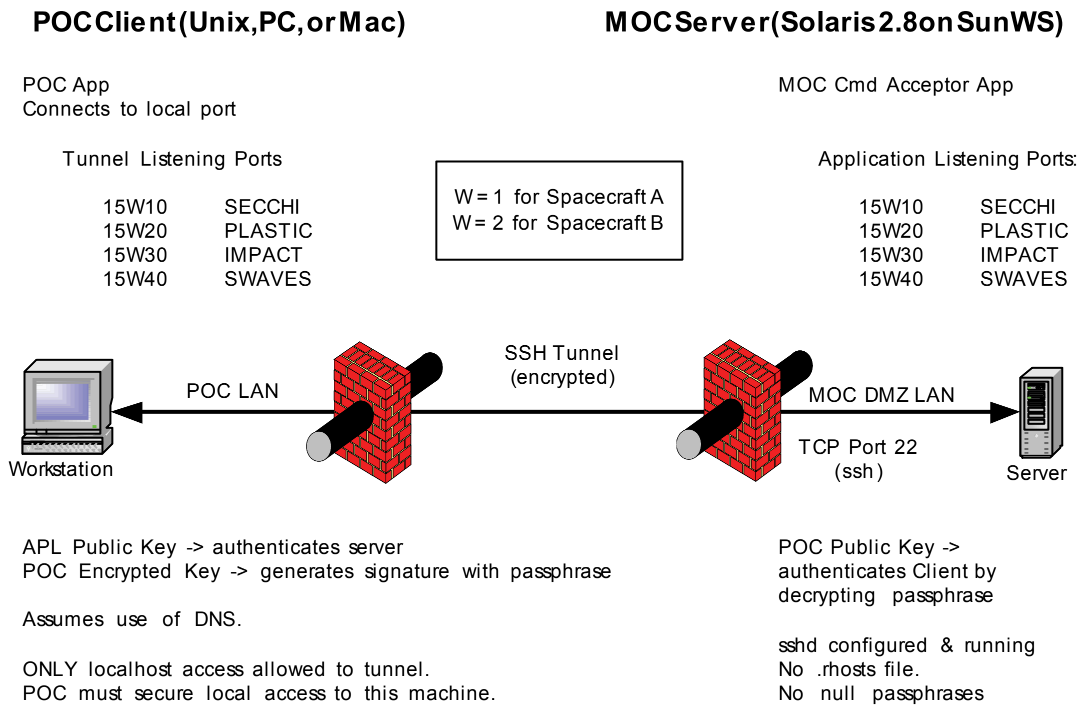


Figure 8. SSH Configuration

2.5 TELEMETRY INTERFACES

2.5.1 TELEMETRY INTERFACE OVERVIEW

Non-SSR dump telemetry data received by the DSMS is forwarded to the MOC in real-time on a best-effort basis. Telemetry forwarded to the MOC in real-time is available in real-time to the POCs and the SSC via TCP/IP streams which transmit telemetry meeting the user's specifications on a best-effort basis.

The complete set of non-SSR dump telemetry and SSR dump telemetry is forwarded to the MOC after the pass is complete. This data is cleaned, merged, and added to an online archive that supports TCP/IP stream playback of the complete data set (instrument data limited to the last 30 days).

L0 files are produced from archived instrument and spacecraft data following ingestion of the post-pass telemetry delivery. Because data for a given spacecraft day might be downlinked on successive days, the "complete" files are regenerated each day for the two preceding days and for the 30th day preceding the current day. This approach provides the majority of the data quickly after it was taken, while ensuring that the POCs and SSC get the most complete set possible. These files are available to the POCs and the SSC via FTP transfer.

Spacecraft subsystem telemetry is available to the POCs and the SSC via all three interfaces mentioned above. JHU/APL will provide telemetry packet specifications for spacecraft subsystem packets to aid the POCs and the SSC in interpreting them.

Real-time and playback telemetry streams and L0 file transfer will normally be conducted over the Internet. In case of Internet outage or excessive latencies, the MOC will support identical protocols (with limited capacity) over modem dial-up service running PPP. Modem communications are considered to be a backup capability and are not sized to replace the Internet path for an extended period. The modem pool is limited to 19,200 bps connections and is shared with the TIMED mission. The MOC will be capable of supporting one or more POCs communicating via modem while one or more POCs communicate via the Internet. The decision to use the modem will be made by the POC.

NOTE: In the description that follows, "spacecraft time" refers to the UTC value stored in the secondary header of a telemetry packet. "Ground receipt time" refers to the DSMS time-of-receipt on the ground, also UTC, and captured in the Ground Receipt Header (GRH) portion of a Supplemented Telemetry Packet (STP).

2.5.2 TELEMETRY DELIVERY TO THE MOC

2.5.2.1 REAL-TIME TELEMETRY

Real-time telemetry is defined as any spacecraft or instrument telemetry that the spacecraft sends down to the DSMS via Virtual Channel 7 (VC7) during a track. During the track, the DSMS will immediately forward the real-time telemetry to the MOC via a UDP-based best-effort, low latency, delivery service.

2.5.2.2 SSR DUMP TELEMETRY

SSR Dump telemetry is defined as any spacecraft or instrument telemetry which was stored on the SSR and subsequently downlinked to the DSMS on Virtual Channel 6 (VC6). Mission operations may configure the observatories such that all or particular portions of instrument and spacecraft telemetry are recorded on the SSR. During a track, the DSMS Central Data Recorder (CDR) will save the Virtual Channel 6 and Channel 7 telemetry to an Intermediate Data Record (IDR) file. The DSMS will forward this file post pass to the MOC via FTP.

2.5.2.3 COMPLETE DELIVERY

A DSMS track is complete when all data transmitted to the DSMS ground station from the STEREO spacecraft is received by the MOC. Therefore, the DSMS track is not finished until the MOC receives the real-time telemetry and the SSR dump telemetry IDR file via FTP from the DSMS CDR. It normally takes some period of time (e.g., 1 hour) for the CDR to receive all the data from the station after the pass, and given the volume of data expected, could take an hour or more to transfer the IDR file to the MOC. Note that the complete and in-order IDR files are the only telemetry source used to populate the archive.

2.5.3 LEVEL 0 TELEMETRY DATA

L0 processing of Spacecraft Telemetry is defined as the minimum level of processing required to develop a data set of "cleaned and merged" telemetry. The processing produces a data set which is suitable for delivery to users of the data who are usually only concerned with the data content, and not the means of spacecraft-to-earth transmission. Any duplicates that occur because data was sent to the ground multiple times or because the data was received at multiple ground stations, shall be removed by L0 processing. Duplicate packets are defined as packets with the same spacecraft time, APID (See Appendix B), packet sequence count, and frame quality flags. The MOC shall not include in L0 data packets from telemetry frames that could not be Turbo-decoded. Consequently, a frame quality of “good” indicates the telemetry frame from which the packet was taken was “successfully” turbo-decoded and passed its CRC check, while a frame quality of “bad” indicates the telemetry frame was “successfully” turbo-decoded but failed its CRC check. The MOC shall not include idle packets (APID 0x7ff) in L0 data.

2.5.4 TELEMETRY REPOSITORIES

2.5.4.1 RAW TELEMETRY ARCHIVE

The MOC maintains a tape-based Raw Telemetry Archive that contains all real-time and SSR dump telemetry received via DSMS from the spacecraft. The archive comprises the unprocessed IDR files as delivered from the DSMS. It is intended to be a safeguard against loss of data due to incorrect processing and is not intended to be accessed on a regular basis.

2.5.4.2 L0 INDEXED ARCHIVE

The MOC maintains two disk-based online indexed archives containing L0 telemetry, one archive for each observatory; associated applications are capable of serving this telemetry to clients via TCP/IP streams. Clients include POCs, SSC, assessment tools, and MOC applications to generate L0 files and other products for the POCs and the SSC. The archive supports a TCP/IP data stream interface, as described in Section 2.5.6.2.

2.5.4.3 L0 FILE ARCHIVE

The MOC maintains a disk-based online file archive of L0 telemetry files. This archive is a component of the SDS.

2.5.5 L0 DATA PROCESSING

2.5.5.1 L0 INDEXED ARCHIVE POPULATION

After the MOC receives from DSMS all data associated with a track, the MOC shall ingest the data into the L0 Indexed Archive. The MOC shall complete ingestion and be able to support telemetry playback as described in Section 2.5.6.2 within 3 hours of receipt of the IDR files from the CDR. The MOC shall maintain spacecraft subsystem data in the archive for the life of mission and instrument data for at least 30 days. Instrument data shall not be deleted from the archive until positive acknowledgment is received from the STEREO Science Center (SSC) that the final instrument telemetry products have been successfully retrieved.

If one or more telemetry packets received during a track has a spacecraft time value more than 3 days earlier than the current date, the MOC shall send the applicable POCs an e-mail message. The e-mail message shall contain the Spacecraft Identifier (SCID) and a list of the Spacecraft year and day of year dates received. The POCs may request a playback of the old Spacecraft Time date telemetry via the Playback Telemetry Interface or (presuming the date is less than 30 days earlier than the current date) wait for the final L0 file to be produced. L0 file production is described in the next section.

2.5.5.2 L0 FILE PRODUCTION

2.5.5.2.1 IN ARCHIVE TIME RANGE

Once a day, and within 24 hours of the completion of a pass, the MOC shall produce L0 telemetry files by instrument and observatory from telemetry in the L0 Indexed Archive. The MOC shall produce files for the current spacecraft day, the preceding spacecraft day, the day before the preceding spacecraft day, and shall produce a “final” L0 file for the day 30 days before the preceding spacecraft day. The intent of the production of files for this set of days is to make as much telemetry as possible available to the POCs early on, while allowing for the fact that some recorder data might be downlinked many days after it was collected.

The MOC shall produce separate L0 files for each instrument/observatory combination; an L0 file for a given instrument will only contain telemetry packets with Application Identifiers (APIDs) in that instrument's assigned range (See Appendix B). There will also be separate files produced that contain L0 data from the spacecraft subsystems for each observatory.

The MOC shall generate one L0 file for each spacecraft day for the IMPACT, PLASTIC, and SWAVES POCs. The MOC shall generate six L0 files for each spacecraft day for the SECCHI POC, with each file containing four hours of the spacecraft day’s telemetry. The MOC shall produce one L0 file for each spacecraft day containing spacecraft subsystem telemetry.

Telemetry packets in the L0 files shall be ordered first by increasing spacecraft time and then by packet sequence count for cases where the times are identical.

L0 filenames shall be as described in Table 7, with the final (30 day) L0 files receiving a ".fin" extension.

Table 7. Level 0 Telemetry Filenames

Descriptor	Definition	Values	Delimiter
instid	6 character Instrument	stereo, impact, plastc, secchi, or swaves	"_"
scid	3 character Spacecraft Identifier	Ahead = "sta" Behind = "stb"	"_"
yyyy_doy *	7 character Spacecraft or Ground Receipt Time	e.g. 2006_309 (GMT)	"_"
#	1 character Decimal File Number	1 = C&DH and G&C, IMPACT, PLASTIC, and SWAVES 1 to 6 = SECCHI	"_"
##	2 character Version Number	01 to 99	"."
extension	3 character File Extension	"ptp" = Level 0 PTP file, "fin" = FINAL version, "oar" = Out of Archive time range	
Example: swaves_stb_2006_144_1_02.ptp			

* The file date field is the Ground Receipt Time for the Out of Archive Time Range Level 0 files and the Spacecraft Time for all other Level 0 files.

2.5.5.2.2 OUT OF ARCHIVE TIME RANGE

The MOC L0 Indexed Archive nominally retains the instrument telemetry data for 30 days. Therefore, if data received during a track has a Spacecraft timestamp which is more than 30 days in the past or has a timestamp in the future, the MOC considers the telemetry to be "Out of Archive Time Range" (OAR) telemetry.

Once a day, the MOC shall generate an L0 file for each instrument/observatory combination containing OAR telemetry received during the previous day, ordered by Ground Receipt Time. (NOTE: The MOC will not create "Empty" OAR files.)

The MOC shall name the OAR L0 files as shown in Table 7 using Ground Receipt Time in the file date field and a file extension of "oar". Then, the MOC shall send the appropriate POCs and SSC an e-mail message which references the OAR L0 files. The e-mail message shall contain the Spacecraft Identifier and the names of the OAR L0 data files.

2.5.5.2.3 L0 FILE RETENTION

The MOC shall retain L0 files in the L0 File Archive for at least 7 days after production and until confirmation is received from the SSC that the files have been transferred to the SAC.

2.5.6 TELEMETRY INTERFACE PROTOCOLS

2.5.6.1 REAL-TIME TELEMETRY INTERFACE

The Real-time Telemetry Interface supports the transmission of real-time telemetry received by the MOC during the DSMS track to the POCs and the SSC. The real-time telemetry shall be sent via a TCP/IP data stream, in the order in which it was received from the DSMS, and without cleaning or merging. Common elements of the real-time and playback interfaces are described in Section 2.5.6.3 and 2.5.6.4.

The MOC will use a non-blocking socket for client connections; consequently, if the POCs do not read data from the TCP/IP stream at the rate it is written (allowing for modest TCP stack buffering), the MOC shall drop the packets that cannot be written to the interface.

There shall not be an indication of the end of the real-time telemetry stream following a pass; the POCs can remain connected between passes or terminate the connection at their discretion. The POCs will get the DSMS schedules (See Section 2.6.8), which will include the AOS and LOS times for the week.

2.5.6.2 PLAYBACK TELEMETRY INTERFACE

The MOC Playback Telemetry interface supports the transmission of the telemetry from the L0 Indexed Archives (one per observatory) to clients via TCP/IP data streams. Common elements of the real-time and playback interfaces are described in Section 2.5.6.3 and 2.5.6.4.

The MOC will use a blocking socket for client connections; consequently, the client (POC or SSC) controls the rate that telemetry packets are transferred and the MOC shall not drop packets. When the stream is at the end of the requested data or at the end of the archive, the MOC shall send the client a Ground Receipt Header with the Data Type Field set to 0, indicating the end of the stream. After all requested data have been received, the POCs are responsible for closing the socket connection to terminate the session.

The telemetry playback may be requested in either ground receipt time or spacecraft time order.

2.5.6.3 MOC / POC TELEMETRY SOCKET INTERFACE PROTOCOL

The Real-Time (process name = 'router') and Playback (process name = 'archive_server') Telemetry Socket Interfaces between the MOC and the POCs or SSC are shown in Figure 9. The MOC processes shall act as servers in the TCP/IP sense and shall establish listening ports for Real-Time and Playback telemetry services.

NOTE: All of the elements of this interface apply equally to the POCs and the SSC. For brevity's sake, only the term "POC" will be used in the description.

Each service (Real-Time and Playback) for each observatory shall support up to 5 simultaneous connections from each POC and the SSC. Each POC shall make a connection request to the MOC on the host and port assigned to that POC to deliver the desired service for the desired observatory. If a POC attempts to open more than the maximum number of allowed connections for a given service, the MOC will not accept the connection. Also, if a POC attempts to establish a TCP/IP connection on a port that is not assigned to it, the MOC shall refuse the connection request. If the port connection is available, the MOC shall accept the connection request from the POC and communications on the connection begins.

Following establishment of the connection, a POC shall send multiple Playback or Real-time Directive Messages to identify the telemetry desired. A summary of the directive messages is given in Table 8. A complete description of the Directive messages, including the directive parameters, can be found in Section 2.5.6.4.

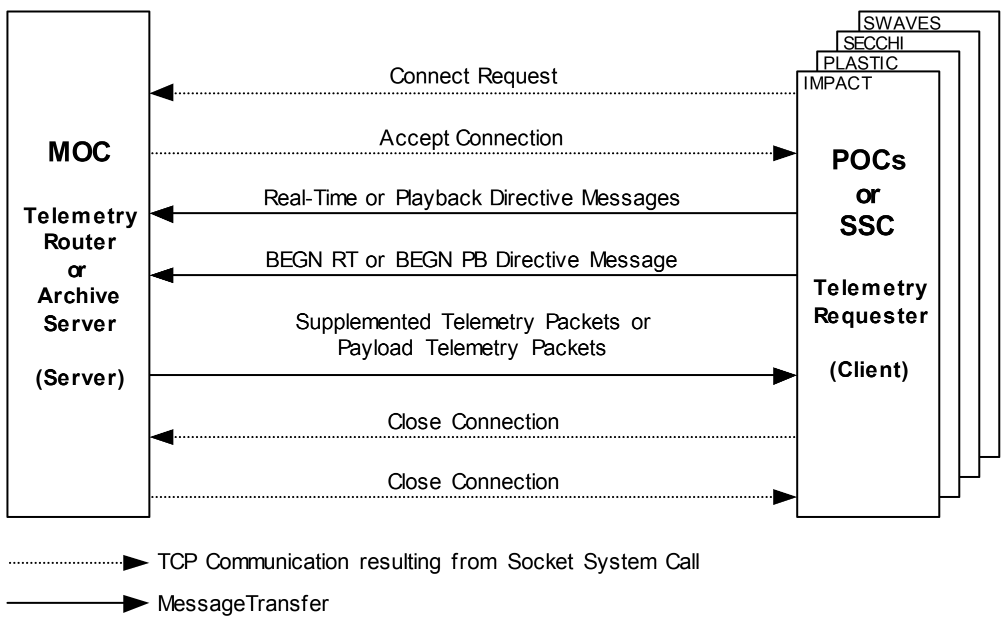


Figure 9. MOC to POC or SSC Telemetry Interface

Table 8. Directive Message Summary

Directive Type	Real-time Streams	Playback Streams	Directive Message Purpose
Select which data is included in the stream	APID	APID	Flow packets with this APID
	EXAPID	EXAPID	Do not flow packets with this APID
	SSYS	SSYS	Flow packets from this subsystem (subsystem ID corresponds to the 4 most significant bits of the APID field)
Select the data sources	FRNT	FRNT	Flow packets that arrived via specific Front Ends or DSN Complexes
	SRCE	SRCE	Flow packets produced by this source. Sources include MOC software processes as well as the Spacecraft.
	STRM	STRM	Flow packets that were collected using a specific MOC hardware & software configuration
	PATH	PATH	Flow packets received by a specific DSN Antenna (This identifier, in combination with the complex ID, uniquely identifies an antenna)
	VCHN	VCHN	Flow packets from this virtual channel
Select extra destination	TLM_HOST	TLM_HOST	Specify a host with which the MOC will initiate a secondary connection to flow telemetry
	TLM_PORT	TLM_PORT	Specify a port on which the MOC will initiate a secondary connection to flow telemetry
Select format	TYPE	TYPE	Select the data format (STP or PTP)
Timestamp Information		STRT	Specify the start time partition & timestamp for data included in the stream
		STOP	Specify the time partition & stop timestamp for data included in the stream
		ORDR	Specify ground receipt or spacecraft timestamp order

The purpose of the directive messages is to specify which telemetry data will be transferred. A POC can select telemetry by several characteristics, including the source of the data, the telemetry data format, and the time range for the playback of the archived telemetry.

After all of the telemetry specification directive messages are sent, the BEGN directive message shall be sent to the MOC to start the transmission of the telemetry data, and the POCs shall wait for the telemetry to be sent on the connection.

The MOC shall read the directive messages from the connection and shall verify that the directives are appropriate and valid. If the directives are invalid, the MOC shall close the connection. If the directives are valid and data is available for transfer, the MOC shall start sending the requested telemetry data on the socket in the requested data format.

If a POC connects to the archive_server and there is no data available matching the POC's specification for the transfer, the MOC shall send an End of Transmission packet (i.e., Ground Receipt Header with the Data Type Field set to 0) and wait for the client (POC) to close the connection.

The MOC shall continue to send the telemetry data to a POC until a POC closes the connection, the stop timestamp is reached (archive_server only), or when no more real-time or playback data is available.

The MOC telemetry server component (router or archive_server) shall close an established connection with a POC ONLY if erroneous directives are received. In all other cases, it is up to the POC to close the connection at its discretion. When the MOC receives an indication that the socket has been closed by the POC, it shall stop sending any additional telemetry data and shall close its end of the connection.

If there is a failure in communications on the connection, the POC shall be responsible for requesting the telemetry data again (which requires disconnection and reconnection). Connection errors might include loss of the socket connection, port in use or not available, or failures on the reads or writes. When an error occurs, the POC shall close the connection, re-establish the connection, and resend the directive messages.

2.5.6.4 REAL-TIME AND PLAYBACK DIRECTIVE MESSAGES

The real-time and playback directive message formats are defined in Table 9. The Directive Messages shall include the directive name, an optional space (" ") or equals sign ("=") delimiter, and the directive parameters, if applicable, and a newline delimiter which terminates the message. The directive name and parameters shall be in UPPERCASE ASCII characters. Only one parameter value can be specified for each directive command message, but the directive command messages may be sent multiple times. The following rules apply to the directive message:

- TLM_PORT and TLM_HOST are optional, but both must be present or both must be absent

- STRT and STOP time are optional for the Playback service and not permitted for the Real-Time service; default sends all data in archive for current day
- BEGN RT or BEGN PB are required and must be the last directive sent
- Any directive sent after the BEGN directive is ignored. The client must disconnect and re-connect in order to change the data requested.

When start/stop times are specified for Playback service you must also indicate which time "partition" should be used. The concept of time partitions is defined by the SPICE system developed by the Jet Propulsion Laboratory's (JPL) Navigation and Ancillary Information Facility (NAIF). Because there can be discontinuities in the progression of the spacecraft clock count (e.g., due to reset), SPICE uses the concept of "partitions" to specify time ranges where the clock monotonically increased. If the clock never jumped, you would only have one partition. The partition start time and stop time represent a period of time over which there were no jumps. For example, suppose the clock counted from 0 to 86400 (a day), and then jumped back 1 second before continuing. Partition 1 would be defined as [0, 86400] and Partition 2 as [86399, <big number>]. Consequently, it is not always enough to know the UTC Stereo Count representation of time - you need to know the partition number as well to unambiguously determine the time.

The list of the real-time and playback directive commands, including the directive names, parameters, description, and default values are defined in Table 10 and Table 11, respectively.

Table 9. Directive Message Format

Directive Message				
Contents	Size (bytes)	Type	Units/Range	Delimiter
Directive Command	Variable	ASCII	See Real-Time and Playback Directive Sheet	Optional Space " " or Equals "="
Directive Command Parameter	Variable	ASCII	See Real-Time and Playback Directive Sheet	Newline "\n"
TOTAL SIZE	Variable			

Table 10. Real-Time Directive Message Types

Real-Time Directive Messages				
Directive	Parameters	Description	Defaults	Required?
APID	Number in octal, hex or decimal (See NOTE 1)	Include Application Process ID in the stream. You can request multiple APIDs, one per directive. You must specify at least one APID or SSYS to receive telemetry. Note that APID=ALL is NOT supported; if you want all APIDs, use SSYS=ALL.	(none)	NO
EXAPID	Number in octal, hex or decimal (See NOTE 1)	Exclude Application Process ID from the stream. You can exclude multiple APIDs, one per directive	(none)	NO
SSYS	decimal number, "ALL"	Include all APIDs that match on the subsystem ID field (4 most significant bits of the APID). You can request multiple subsystems, one per directive. You must specify at least one APID or SSYS to receive STP or PTP. For all subsystems use SSYS=ALL.	ALL	NO
FRNT	decimal number, "ALL"	You can request multiple FRNTs, one per directive. For normal operations, this code corresponds to a DSN complex (e.g., Goldstone). You can get all of the front ends by using the keyword ALL. The ALL option will send duplicate streams for a source, if there are multiple input streams from the same source.	ALL	NO
SRCE	decimal number, "ALL"	You can request multiple sources, one per directive. You can get all of the sources by specifying "ALL". POCs should normally specify "ALL".	ALL	NO
STRM	decimal number, "ALL"	You can get all of the Streams by specifying "ALL". POCS should normally specify "ALL"	ALL	NO
PATH	decimal number, "ALL"	Identifies DSN Ground Station within a complex. You can get data from all providing stations within a complex by specifying "ALL". POCS should normally specify "ALL"	ALL	NO
VCHN	"0","6","7", "ALL"	You can request multiple VCHNs, one per directive.	ALL	NO
TLM_HOST	decimal number ddd.ddd.ddd.ddd	Host IP number for remote socket connection for secondary telemetry stream destination - if not the same as first socket (host names not allowed) Directives are sent on the primary socket connection. Data flows on the secondary socket connection. The primary socket must remain open until all data is received on the secondary connection.	same as IP of first socket connection	NO
TLM_PORT	decimal number	Port number for remote socket connection for secondary telemetry stream destination (required if second socket requested)	(none)	NO
TYPE	"STP" or "PTP"	Specify whether to get Supplemented Telemetry Packets or Payload Telemetry Packets. Only one type may be specified.	STP	NO
BEGN	RT	Start to Send Real-Time Telemetry Data	n/a	YES

NOTE 1:The expected form of numeric parameters is that of a decimal constant, octal constant or hexadecimal constant, any of which may be preceded by a + or - sign. A decimal constant begins with a non-zero digit, and consists of a sequence of decimal digits. An octal constant consists of the prefix 0 optionally followed by a sequence of the digits 0 to 7 only. A hexadecimal constant consists of the prefix 0x or 0X followed by a sequence of the decimal digits and letters a (or A) to f (or F) with values 10 to 15 respectively.

Table 11. Playback Directive Message Types

Playback Directive Messages				
Directive	Parameters	Description	Defaults	Required?
APID	Number in octal, hex or decimal (See NOTE 1)	Include Application Process ID in the stream. You can request multiple APIDs, one per directive. You must specify at least one APID or SSYS to receive telemetry. Note that APID=ALL is NOT supported; if you want all APIDs, use SSYS=ALL.	(none)	NO
EXAPID	Number in octal, hex or decimal (See NOTE 1)	Exclude the specified Application Process ID from the stream. You can request multiple APIDs for exclusion, one per directive.	(none)	NO
SSYS	decimal number, "ALL"	Include all APIDs that match on the subsystem ID field (4 most significant bits of the APID). You can request multiple subsystems, one per directive. You must specify at least one APID or SSYS to receive STP or PTP. For all subsystems use SSYS=ALL.	ALL	NO
FRNT	decimal number, "ALL"	You can request multiple FRNTs, one per directive. For normal operations, this code corresponds to a DSN complex (e.g., Goldstone). You can get all of the front ends by using the keyword ALL. The ALL option will send duplicate streams for a source, if there are multiple input streams from the same source.	ALL	NO
SRCE	decimal number, "ALL"	You can request multiple sources, one per directive. You can get all of the sources by specifying "ALL". POCs should normally specify "ALL".	ALL	NO
STRM	decimal number, "ALL"	You can get all of the Streams by specifying "ALL". POCS should normally specify "ALL"	ALL	NO
PATH	decimal number, "ALL"	Identifies DSN Ground Station within a complex. You can get data from all providing stations within a complex by specifying "ALL". POCS should normally specify "ALL"	ALL	NO
VCHN	"0", "6", "7", "ALL"	Include the specified Virtual Channel ID in the stream You can request multiple VCHNs, one per directive.	ALL	NO
TLM_HOST	decimal number ddd.ddd.ddd.ddd	Host IP number for remote socket connection for secondary telemetry stream destination - if not the same as first socket (host names not allowed). Directives are sent on the primary socket connection. Data flows on the secondary socket connection. The primary socket must remain open until all data is received on the secondary connection.	same as IP of first socket connection	NO
TLM_PORT	decimal number	Port number for remote socket connection for secondary telemetry stream destination (required if second socket requested)	(none)	NO
TYPE	"STP" or "PTP"	Specify whether to get Supplemented Telemetry Packets or Payload Telemetry Packets. Only one type may be specified.	STP	NO
STRT	N / yyyy doy hh:mm:ss	Time Partition (N) and start time – must be before time of last data in archive	start of current UTC day	NO
STOP	N / yyyy doy hh:mm:ss	Time Partition (N) and stop time - if stop time exceeds the time of the last data in the archive, the server will wait for new data to arrive.	time of last data in archive	NO
ORDR	"SC", "GR"	Spacecraft time or Ground Receipt time order. Prior to launch only ground receipt time ordering will be available.	GR	NO
BEGN	PB	Start to Send Playback Telemetry Data	n/a	YES

NOTE 1:The expected form of numeric parameters is that of a decimal constant, octal constant or hexadecimal constant. A decimal constant begins with a non-zero digit, and consists of a sequence of decimal digits. An octal constant consists of the prefix 0 optionally followed by a sequence of the digits 0 to 7 only. A hexadecimal constant consists of the prefix 0x or 0X followed by a sequence of the decimal digits and letters a (or A) to f (or F) with values 10 to 15 respectively.

2.5.6.5 SSC / MOC SPACE WEATHER DATA INTERFACE

The SSC shall retrieve the Space Weather data from the MOC either via a real-time socket connection during the DSMS track (Section 2.5.6.1) or via a playback socket connection (Section 2.5.6.2).

2.5.6.6 LEVEL 0 FILE FTP INTERFACE

The POCs shall retrieve the L0 data files from the MOC L0 file archive via FTP. The SSC shall retrieve the final versions of the L0 files (those created for spacecraft days 30 days prior to the current day) from the MOC L0 file archive via FTP for storage in the SSC Long Term archive.

The MOC shall provide a list of the available L0 telemetry file names on an HTTP web page. The Level 0 telemetry filename fields are shown in Table 7.

The POCs shall retrieve the L0 files within 7 days of their production, and may retrieve the final version of the L0 files. The POCs shall retrieve the OAR L0 files within 7 days of their production. Automatic FTP of all of the L0 files shall be allowed.

2.5.6.7 SSC/ MOC LONG TERM ARCHIVE INTERFACE

The SSC shall store all L0 files in the SAC Long Term Archive for the life of the mission. On a daily basis, the SSC shall retrieve the final version of the L0 files from the MOC via FTP. Also, when the SSC receives e-mail notification of the OAR L0 files, the SSC shall retrieve the files from the MOC via FTP. If the POCs need to retrieve an L0 telemetry data file after deletion from the MOC, the SSC shall make the file available for the POCs.

Each day, the SSC shall generate a Archived Product List (APL) that indicates which files were transferred to the SAC that day. The file names for the APL files are shown in Table 12. The APL shall be a text file that contains a list of the names and size in bytes of the files retrieved by the SSC. Each archived filename entry in the APL shall be on a separate line in the APL, and the name and size of the archived files shall be separated by at least one white space (tab or space) character.

Table 12. Archived Product List Filenames

Descriptor	Definition	Example(s)
SCID	3 character spacecraft identifier	Ahead = “STA” (hex) Behind = “STB” (hex)
file date	GMT yyyy_doy	2006_143
####	Padded 4 digit revision ID	0001
e.g., stb_2006_143_0001.apl		

Periodically, the MOC shall retrieve the APL files from the SSC via FTP. The MOC shall use the APL to verify that the L0 telemetry files are in the SAC and that the files were not corrupted by the transfer prior to deleting telemetry in the L0 Archive. The MOC shall consider a difference in file size to signal that the file was corrupted. The MOC shall not delete the final version of the L0 file or the OAR L0 file until it has been confirmed the files have been successfully stored in the SAC. Also, the MOC shall not delete the instrument telemetry from the L0 Archive until the associated L0 file is stored in the SAC.

2.5.7 TELEMETRY MESSAGE FORMATS

2.5.7.1 SUPPLEMENTED TELEMETRY PACKETS

The STP, primary frame header, and secondary frame header formats are shown in Figure 10, Figure 11, and Figure 12, respectively. The STP attributes, including the data size, type, and values for each field in the message are defined in Table 13. The format of the CCSDS telemetry packet within the STP is described in Section 2.5.7.3. The Ground Receipt Header, which is also part of the STP, is described in Section 2.5.7.4.

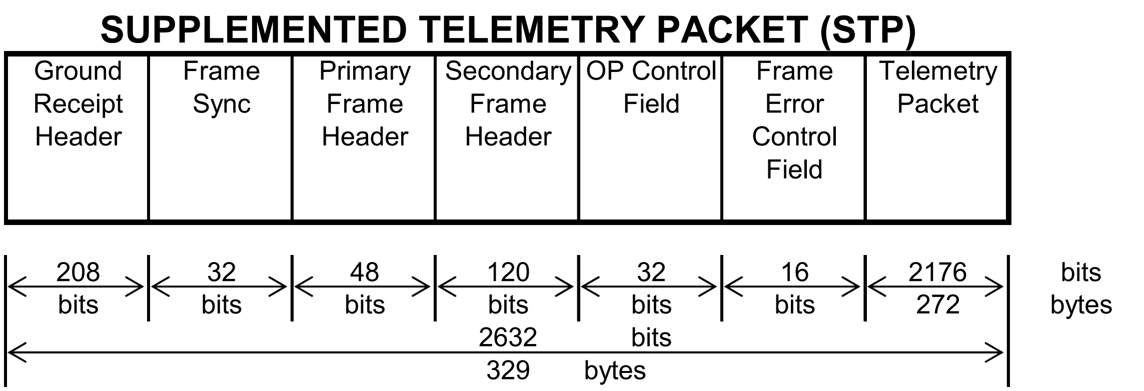


Figure 10. Supplemented Telemetry Packet Data Structure Diagram

PRIMARY FRAME HEADER

Version Number	Identification Field			Master Channel Frame Count	Virtual Channel Frame Count	Data Field Status				
	Spacecraft ID	Virtual Channel ID	Op Control Field			Secondary Header Flag	Sync Flag	Packet Order Flag	Segment Length ID	First Header Pointer
"00"	Ahead = 0XEA = "0011101010" Behind = 0xEB = "0011101011"	0 = "000" 6 = "110" 7 = "111"	Normally "1"	Counter Decimal 0 to 255	Counter Decimal 0 to 255	1 = present 0 = not present	Normally "0"	Normally "0"	Normally "11"	Decimal Range 0 to 2047
2 bits	10 bits	3 bits	1 bits	8 bits	8 bits	1 bits	1 bits	1 bits	2 bits	11 bits

48 bits / 6 bytes

Figure 11. Telemetry Frame Primary Header Data Structure Diagram

SECONDARY FRAME HEADER

Identification Field		Data Field		
Version Number	Header Length	Spacecraft MET TIme Seconds	Spacecraft MET Time Subsecs	Spares
"00"	"001110"	Rollover in 136 years	4 msec counts	
2 bits	6 bits	32 bits	8 bits	72 bits

120 bits / 15 bytes

Figure 12. Telemetry Frame Secondary Header Data Structure Diagram

Table 13. Supplemented Telemetry Packet Format

SUPPLEMENTED TELEMETRY PACKET (STP)				
Contents	Size (bits)	Size (bytes)	Type	Units/Range
Ground Receipt Header	208	26	Varies	See Ground Header Sheet
Frame Sync	32	4	Numeric	1ACF FC1D
TELEMETRY TRANSFER FRAME PRIMARY HEADER				
Version Number	2	0.250	Binary	"00"
IDENTIFICATION FIELD				
Spacecraft ID	10	1.250	Binary	Ahead = 0XEA = "0011101010" Behind = 0xEB = "0011101011"
Virtual Channel ID	3	0.375	Binary	0 = "000" 6 = "110" 7 = "111"
Operational Control Field	1	0.125	Binary	Normally "1"
Master Channel Frame Count	8	1.000	Binary	Counter Decimal 0 to 255
Virtual Channel Frame Count	8	1.000	Binary	Counter Decimal 0 to 255
DATA FIELD STATUS				
Secondary Header Flag	1	0.125	Binary	1 = secondary header present 0 = secondary header not present
Sync Flag	1	0.125	Binary	Normally "0"
Packet Order Flag	1	0.125	Binary	Normally "0"
Segment Length ID	2	0.250	Binary	Normally "11"
First Header Pointer	11	1.375	Binary	Decimal Range 0 to 2047
SECONDARY FRAME HEADER				
IDENTIFICATION FIELD				
Version Number	2	0.250	Binary	"00"
Secondary Header Length	6	0.750	Binary	Secondary Header Length - 1 = Decimal 14 = "1110"
DATA FIELDS				
Spacecraft MET Time Seconds	32	4.000	Binary	Rollover in 136 years
Spacecraft MET Time Subsecs	8	1.000	Binary	4 msec counts
Spares	72	9.000	Binary	Spares
Operational Control Field	32	4.000	Numeric	Format in CCSDS document 202.0-B-3
Frame Error Control Field	16	2.000	Numeric	Encoding Method in CCSDS document 102.0-B-4
Telemetry Packet	2176	272	Varies	See TLM Packet Sheet
TOTAL SIZE (bits & bytes)	**2632**	**329**		

2.5.7.2 PAYLOAD TELEMETRY PACKETS

The Payload Telemetry Packet structure is shown in Figure 13. The PTP attributes, including the data size, type, and units for each component are defined in Table 14. The format of the CCSDS telemetry packet within the payload telemetry packet is described in Section 2.5.7.3. The Ground Receipt Header, which is also part of the payload telemetry packet, is described in Section 2.5.7.4.

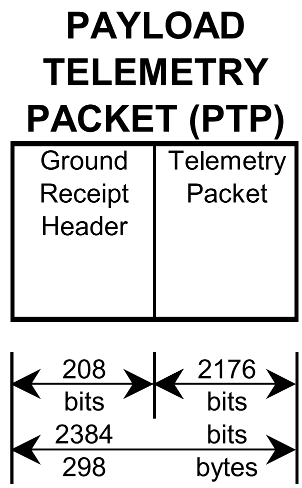


Figure 13. Payload Telemetry Packet Data Structure Diagram

Table 14. Payload Telemetry Packet Format

PAYLOAD TELEMETRY PACKET (PTP)				
Contents	Size (bits)	Size(bytes)	Type	Units/Range
Ground Receipt Header	208	26	Varies	See Ground Header Sheet
Telemetry Packet	2176	272	Varies	See TLM Packet Sheet
TOTAL SIZE (Bits & Bytes)	**2384**	**298**		

2.5.7.3 CCSDS TELEMETRY PACKET

The CCSDS telemetry packet structure is shown in Figure 14. The CCSDS telemetry packet format, including the data size, type, and values for each field in the message are defined in Table 15.

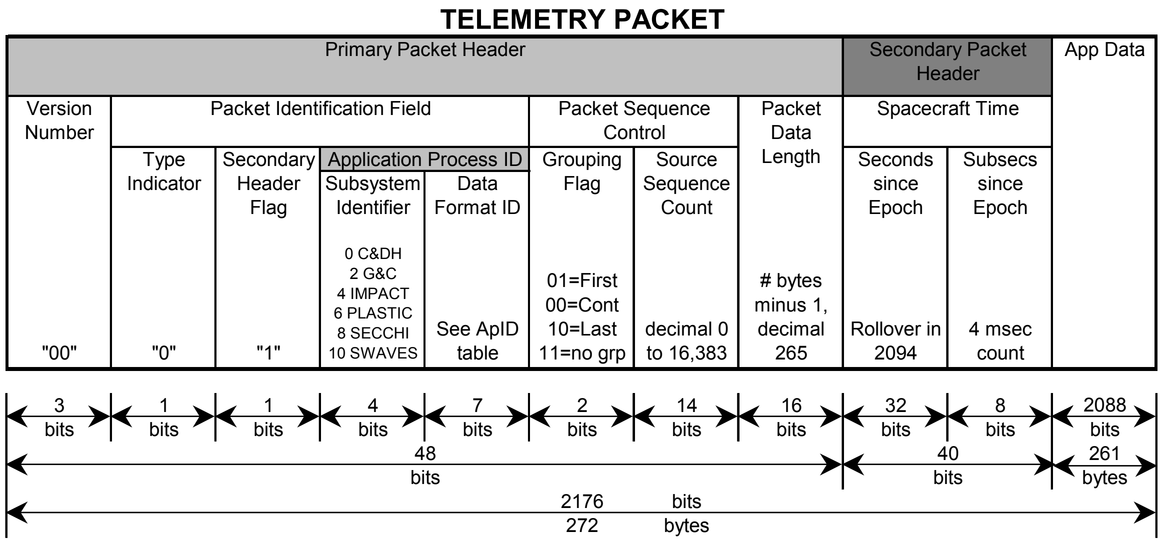


Figure 14. CCSDS Telemetry Packet Data Structure Diagram

Table 15. CCSDS Telemetry Packet Format

CCSDS Telemetry Packet				
Contents	Size (bits)	Size (bytes)	Type	Units/Range
PRIMARY PACKET HEADER				
Version Number	3	0.375	Binary	"00"
PACKET IDENTIFICATION FIELD				
Type Indicator	1	0.125	Binary	"0" designates a telemetry packet
Secondary Header Flag	1	0.125	Binary	0 = No secondary header 1 = Secondary Header Present
APPLICATION PROCESS IDENTIFIER				
Subsystem ID	4	0.500	Binary	0 = 0000 = C&DH 2 = 0010 = G&C 4 = 0100 = IMPACT 6 = 0110 = PLASTIC 8 = 1000 = SECCHI 10 = 1010 = SWAVES
Data Format ID	7	0.875	Binary	See ApID table
PACKET SEQUENCE CONTROL				
Grouping Flag	2	0.250	Binary	01 = First Packet 00 = Cont. Packet 10 = Last Packet 11 = No grouping
Source Sequence Count	14	1.750	Binary	Decimal range = 0 to 16,383 To ensure delivery order, increment this counter
Packet Data Length	16	2.000	Binary	Number of bytes in Secondary Header Fields + Application Data Field **minus 1, For STEREO = (261 + 5) - 1 = decimal 265 = binary 00000000 10001001**
SECONDARY PACKET HEADER				
SPACECRAFT TIME				
Seconds since Epoch	32	4.000	Binary	Seconds since Epoch Jan 01, 1958 00:00:00 UTC, Rollover in 2094
Subseconds	8	1.000	Binary	Subseconds (1/256)
Application Data	2088	261	Variable	Telemetry application data
TOTAL SIZE (bits & BYTES)	**2176**	**272**		

2.5.7.4 GROUND RECEIPT HEADER

The Ground Receipt Header is part of the Supplemented Telemetry Packet and Payload Telemetry Packet. The Ground Receipt Header structure is shown in Figures 15 and 16. The Ground Receipt Header attributes, including the data size, type, and values for each field in the message are defined in Tables 16 and 17.

GROUND RECEIPT HEADER

Size	Data Type	GRT Usage	GRH Version ID	SC ID	Ground Receipt Time			Spare Bits	Frame Quality Flag	Archive Flag
					Epoch Day	Millisecs of Day	Extented Re-solution			
decimal 208	0 = EOT 2 = STP 3 = PTP	010xxxxx		Ahead = 0XEA Behind = 0xEB	Days since Jan 1, 1958		Tenths of microsec onds		0 = Data Suspect 1 = Data Correct	0 = Do Not Archive 1 = Archive
16 bits	8 bits	8 bits	8 bits	16 bits	16 bits	32 bits	16 bits	6 bits	1 bits	1 bits

Total: 208 bits, 26 bytes

Figure 15. Ground Receipt Header Data Structure Diagram (1 of 2)

Table 16. Ground Receipt Header Format (1 of 2)

GROUND RECEIPT HEADER (1 of 2)				
Contents	Size (bits)	Size (bytes)	Type	Units/Range
Size	16	2	Binary	Size of this object including headers in bytes, unsigned integer in MSB first order (max = 65535)
Data Type	8	1	Binary	type of data object 0 = "00000000" = EOT 2 = "00000010" = STP 3 = "00000011" = PTP
GRT Usage	8	1	Binary	Defines how ERT/GRT time should be interpreted. Expected value is 010xxxxx. 1xxxxxxx indicates ERT is invalid
GRH Version ID	8	1	Binary	Version Identifier associated with this GRH format
Spacecraft ID	16	2	Binary	Ahead = 0XEA = "0000 0000 1110 1010" Behind = 0xEB = "0000 0000 1110 1011"
Ground Receipt TIme (GRT) Epoch Day	16	2	Binary	Days since Jan 1, 1958; Range = 0-65,535; Binary Unsigned Integer
GRT milliseconds of day	32	4	Binary	Milliseconds-of-day since midnight UTC; Range = 0 to 86,400,000 (allows for one leap second) Binary unsigned integer
GRT extended resolution	16	2	Binary	Tenths of microseconds since the millisecond; range 0-9,999.
spare bits	6	0.750	Binary	
Frame Quality Flag	1	0.125	Binary	0 = Data Suspect 1 = Data Correct
Archive Flag	1	0.125	Binary	0 = Do Not Archive 1 = Archive

GROUND RECEIPT HEADER (continued)

Frame Source				Coding Fields			Spare Bits
APL MOC Use (Stream ID)	Source Process ID	Front End ID	DSN Antenna ID	Encoding Type	Decode Quality	Project Specific Flags	
8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	24 bits

Figure 16. Ground Receipt Header Data Structure Diagram (2 of 2)

Table 17. Ground Receipt Header Format (2 of 2)

GROUND RECEIPT HEADER (2 of 2)				
Contents	Size (bits)	Size (bytes)	Type	Units/Range
APL MOC Use (Stream ID)	8	1	Binary	
Frame Source Process ID	8	1	Binary	Frame Source Type and Frame Source Index
Front-end Identifier	8	1	Binary	"Front End" identifiers map either to APL MOC, Front Ends, or DSN facilities
Frame Source: DSN Antenna ID	8	1	Binary	DSN Station Identifier
Encoding Type	8	1	Binary	Bit 0 is MSB Bit 0 = 1 No encoding Bit 1 = 1 RS encoding Bit 2 = 1 Convolutional encoding Bit 3 = 1 CRC used Bit 4 = 1 Turbo used Bits 5 - 7 unused
Decode Quality	8	1	Binary	Bit 0 is MSB Bits 0 - 3, 7 unused Bit 4 Decoder success flag; passed=1 Bit 5 CRC check status flag; passed=1 Bit 6 Output type;decoded data=0
Project Specific Flags	8	1	Binary	Not used by STEREO
spare bits	24	3	Binary	Not used by STEREO
TOTAL SIZE (bits & bytes)	**208**	**26**		

2.5.8 EMULATOR, I&T MOC, AND MOC TELEMETRY INTERFACE DIFFERENCES

During integration and test, spacecraft time may not be unique so the best option for telemetry playback order is by ground receipt time. Otherwise, the user will receive all the data for multiple simulations performed at the same spacecraft time. During mission operations, the preferred option for telemetry playback is usually by spacecraft time.

During instrument development, while using the emulator for instrument command and telemetry, only real-time telemetry shall be available. Also, no real-time directives shall be applicable for the Emulator. No L0 data processing of the telemetry shall be performed by the emulator. The emulator shall directly forward all telemetry packets received from the instrument Data Processing Unit as PTPs.

2.6 SUPPORT PRODUCT INTERFACE

A complete description of all STEREO data products including specific file names, formats of the data product files, purpose of product, trigger for generation of the product, and samples of each product shall be provided in the STEREO MOC Data Products Document (Reference 13). The data products generated by the APL MOC shall be available via FTP through a web browser. Below is a summary description of each product currently envisioned to be generated by the APL MOC for retrieval by the instrument POCs and by SSC for storage in the SAC.

2.6.1 ARCHIVED PRODUCT LIST (APL)

The SSC shall generate an Archived Product List (APL) that indicates which L0 Telemetry files were transferred to the SAC that day. This product is used by the MOC to determine when to delete the L0 files and the associated telemetry from its L0 Archive after 30 days. The SSC is the long term archive of all data for the STEREO mission.

2.6.2 AS RUN TRACK PLAN (ATP)

The MOC shall provide a time ordered history of significant MOC events which occurred during the track. The MOC shall also provide information on any anomalies during the track such as data retransmissions and rejected command loads. This report will also contain a list of the instrument command messages sent to the spacecraft during the track as well as any messages which were not sent but scheduled to be sent. All the entries shall have a time field and a description of the activity associated to them.

2.6.3 ATTITUDE HISTORY (AH)

The MOC shall provide a history of both observatories' attitudes. The attitude data shall contain information on the rotational motion and pointing stability of the observatory relative to an inertial reference system.

2.6.4 COMMAND HISTORY (CH)

Mission operations shall keep a time ordered history of observatory commands for each observatory. This history shall include the command, time sent to the observatory, verification history and time actuated.

2.6.5 CONVERTED S/C HOUSEKEEPING (CSH)

The MOC shall generate tables of agreed upon spacecraft housekeeping parameters decoded into engineering units. This data product shall be generated on a daily basis covering a 24 hour period. This data product shall be available within 24 hours of the end of the day for which it was generated.

2.6.6 DAILY TIMELINE (DT)

The MOC shall generate a graphical daily timeline each day depicting the events and activities on both S/C. This timeline will include DSN tracks, antenna, data rate, encoding, SSR playback times, data volume, S/C uplink times, S/C maintenance events and durations, and instrument uplink times.

2.6.7 DATA LOSS LOG (DLL)

The Data Loss Log shall identify any gaps in the data dumped from the spacecraft. It shall also indicate the status of any previous gaps which may have been filled. If the daily data set is as complete as it will ever be, either no gaps or with gaps that cannot be filled, the data set will be marked as such. Once a data set is marked as complete, it shall no longer appear on this report. Each data set shall cover a 24 hour period. The Data Loss Log shall be generated on a daily basis.

2.6.8 DSN SCHEDULE (DS)

The MOC shall keep and provide a log of past and upcoming DSN tracks. This log shall contain information describing the scheduled DSN track times (AOS and LOS), supporting station and the designated antenna service identified for each observatory. This is the deconflicted DSN schedule produced by the RAPT covering the next six to eight weeks for use by the flight projects and the DSN stations.

2.6.9 EPHEMERIDES (EPM)

The observatory ephemerides shall contain translational motion of the observatory relative to an inertial reference system. This data shall include both the history of the observatory's motion as well as trajectory predictions through end-of-life. This shall include the position and velocity of the observatory within TBD time intervals.

2.6.10 EPHEMERIDES TO DSN (XSP)

The STEREO ephemeris shall be provided to JPL for DSN antenna pointing and viewperiod generation. This ephemeris is generated by the FDF, renamed to the specified DSN filename by the MOC, and ftp'd to the NSS at JPL.

2.6.11 HGA GIMBAL ANGLES (HGA)

The MOC shall provide HGA gimbal angle data to the FDF. The HGA angle will correspond to a pre-defined S/C area (Sun facing cross section).

2.6.12 INSTRUMENT EVENT SCHEDULE (IES)

Each POC shall generate an Instrument Event Schedule (IES) that lists all respective instrument events that are planned for both observatories that require external coordination with the MOC, other instruments, or the other S/C. This product is used by the MOC for advance planning of S/C activities and resources. One event is listed per line.

2.6.13 L0 FILES

The L0 files produced by the MOC are completely described in Section 2.5.5

2.6.14 MANEUVER RADIAL COMPONENT (MRC)

The FDF shall provide a report on the achieved radial delta V component (the component along the line-of-sight from the ground station to the observatory) as measured from the Doppler tracking data.

2.6.15 MANEUVER PLAN (MP)

The MOC shall provide to the FDF information on each orbit maneuver. It should include at a minimum the following:

- Maneuver execution time
- Burn duration
- Predicted delta-V vector
- Pre- and post maneuver mass
- Predicted pre- and post maneuver state vectors
- Predicted accelerations through the maneuver

2.6.16 MOC STATUS REPORT (MSR)

The MOC shall provide a high level weekly status report on the all S/C and MOC activities from the previous week. This data product will include information on all S/C and ground system configuration changes and mission significant anomalies, e.g., unsuccessful DSN tracks, S/C anomalies, and MOC outages.

2.6.17 MOMENTUM DUMP PREDICTION (DP)

The MOC shall provide to the FDF information on each predicted momentum dump for modeling residual delta V in the trajectory.

2.6.18 POST MANEUVER REPORT (PMR)

The MOC shall provide to the FDF information on each completed orbit maneuver. It should include at a minimum the following:

- Actual maneuver execution time
- Actual burn duration
- Reconstruction of predicted delta-V vector
- Post maneuver mass
- Reconstruction of predicted pre- and post maneuver state vectors
- Actual accelerations through the maneuver

2.6.19 POST MOMENTUM DUMP SUMMARY (PDS)

The MOC shall provide to the FDF information on the post-momentum dump and/or measured momentum dump delta-V vector for modeling residual delta V in the trajectory. It should include at a minimum the following:

- Actual execution time
- Actual burn duration
- Delta-V vector (in J2000 or heliocentric)
- Fuel usage and observatory mass update
- Actual accelerations through the momentum dump

2.6.20 POST MOMENTUM DUMP THRUST HISTORY (PDT)

The MOC shall provide to the FDF information on the post-momentum dump and/or measured momentum dump delta-V vector for modeling residual delta V in the trajectory. It will consist of a time history of impulse of each thruster firing.

2.6.21 SECCHI GUIDE TELESCOPE FILTERED DATA (GT)

To be specified by the SECCHI Team. Described as Guide Telescope data that was used by the G&C. SECCHI has asked for this product three times during the mission: once before SECCHI commissioning, and two times during heliocentric orbit operations.

2.6.22 SEQUENCE OF EVENTS (SOE)

The MOC shall provide a keyword file to the NOCC specifying the expected S/C uplink and downlink telemetry configuration and the required DSN station configuration for each track.

2.6.23 STATE VECTORS (SV)

The observatory state vectors shall contain the vector epoch, Cartesian position and velocity, observatory mass, average cross-sectional area, and solar radiation coefficient. This data product is produced and retrieved from the FDF with the same frequency as the ephemerides data product.

2.6.24 STORED COMMAND BUFFER (SCB)

The Stored Command Buffer data product shall reflect the contents of the instrument stored command buffer on the spacecraft following each instrument command upload. The report shall reflect the results of a dump of the buffer and the time associated with the dump. This report shall be available on a daily basis.

2.6.25 TELECOMMAND PACKET HISTORY (PH)

Mission operations shall keep a time ordered history of observatory telecommand packets executed and the resolution on each observatory.

2.6.26 TELEMETRY DICTIONARY (TD)

The telemetry dictionary contains all the necessary information to describe each S/C generated telemetry (only, no instrument generated telemetry) in each APID.

2.6.27 TIME HISTORY (TH)

The MOC shall provide a history of the STEREO spacecraft clock drift rate. This product shall characterize the spacecraft clock accuracy for the mission.

2.6.28 TRACK PLAN (TP)

The MOC shall provide a time ordered plan of significant MOC events for each scheduled track.

2.6.29 DSN VIEWPERIOD (VPD)

The DSN shall provide station site view reports for those DSN stations that provide support for STEREO.

2.6.30 FDF VIEWPERIOD (VPF)

The FDF shall provide station site view reports for those DSN stations that provide support for STEREO. These reports will be delivered in a long term report (8 weeks out for all DSN antennas) and a short term report (1 week out for all DSN antennas).

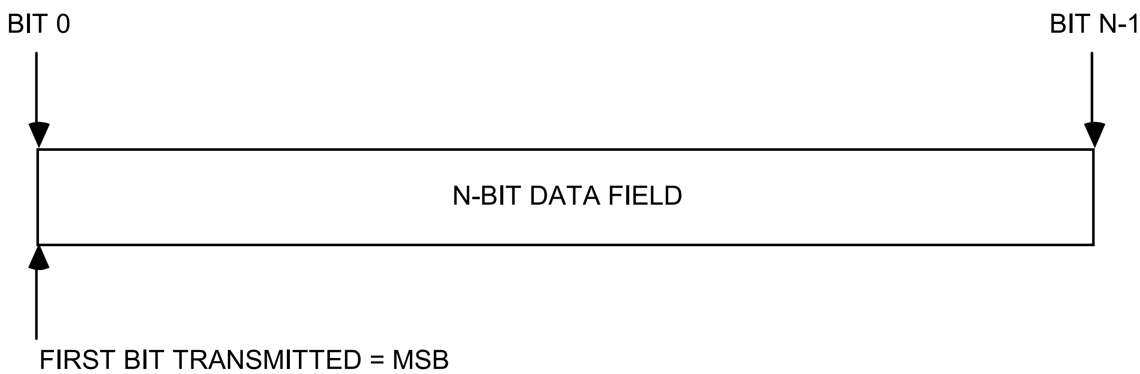
2.6.31 WEEKLY SCHEDULE (WS)

The MOC shall generate a schedule of a week in duration listing the activities and events on both observatories. The MOC week begins on Monday 0000Z to Sunday at 2359Z (same as the DSN). The Weekly Schedule is generated ten days in advance of the scheduled time period.

APPENDIX A. CONVENTIONS

A.1 Bit Ordering & Definition

To identify each bit in an N-BIT FIELD the first bit in the field to be transferred (i.e. the most left justified when drawing a figure) is defined to be "Bit 0"; the following bit is defined to be "Bit 1" and so on up to "Bit N-1". When the field is used to express a binary value (such as a counter), the MOST SIGNIFICANT BIT shall be the first bit of the field, i.e., "Bit 0" (see Figure). (Definition from CCSDS Blue Books at www.ccsds.org)



A.2 Byte Size

One Byte = 8 Bits.

A.3 Data Fields

Data structures are divided into fields, where a field is a sequence of bits. Fields are identified by specifying the starting and ending bits of the field (e.g., "byte 2, bit 5, through byte 2, bit 8," identifies the right 4 bits in byte 2). For fields that cross byte boundaries, bit 8 of byte M is more significant than, and is immediately followed by, bit 1 of byte M+1. A field may be divided into subfields in a similar manner. (JPL Document 820-013 TLM-3-29 Telemetry Standard Formatted Data Unit (SFDU) Interface)

A.4 Binary Unsigned Integers

Binary unsigned integer. A non-negative integer is expressed in binary, using all bits of the field as necessary. Negative quantities cannot be expressed. For an *n*-bit field, the range of values that can be represented is from 0 to $2^n - 1$. (JPL Document 820-013 TLM-3-29 Telemetry Standard Formatted Data Unit (SFDU) Interface)

A.5 ASCII

Restricted ASCII. Each decimal digit of an integer can be expressed by its corresponding RA code. The field must be an integral number of octets in length. For multi-digit fields, the first octet of the field contains the most significant digit, the second octet contains the next most significant digit, and so on. If the number of digits is less than the number of octets in the field, leading zeroes are used to fill the field. Negative quantities cannot be expressed. For an *n*-octet field, the range of values that can be represented is from 0 to 10^n–1. (JPL Document 820-013 TLM-3-29 Telemetry Standard Formatted Data Unit (SFDU) Interface)

APPENDIX B. APPLICATION IDENTIFIERS (APID)

Data Type	Subsystem Identifier (Decimal)	Data Format Identifier (Decimal)	APID Range (Decimal)	APID Range (HEX)	Sub Id 4MS bits (Binary)	7LS bits - Id Start (Binary)	7LS bits - Id Stop (Binary)
Spacecraft Subsystem	0	0 - 63	0 - 63	000 - 03F	0000	000 0000	011 1111
Spacecraft Subsystem	0	64 - 127	64 - 127	040 - 07F	0000	100 0000	111 1111
Spacecraft Subsystem	1	0 - 63	128 - 191	080 - 0BF	0001	000 0000	011 1111
Spacecraft Subsystem	1	64 - 127	192 - 255	0C0 - 0FF	0001	100 0000	111 1111
Spacecraft Subsystem	2	0 - 63	256 - 319	100 - 13F	0010	000 0000	011 1111
Spacecraft Subsystem	2	64 - 127	320 - 383	140 - 17F	0010	100 0000	111 1111
Spacecraft Subsystem	3	0 - 63	384 - 447	180 - 1BF	0011	000 0000	011 1111
Spacecraft Subsystem	3	64 - 127	448 - 511	1C0 - 1FF	0011	100 0000	111 1111
IMPACT	**4**	**0 - 111**	**512 - 623**	**200 - 26F**	**0100**	**000 0000**	**110 1111**
IMPACT Space Weather	**4**	**112 - 127**	**624 - 639**	**270 - 27F**	**0100**	**111 0000**	**111 1111**
Reserved	5	0 - 127	640 - 767	280 - 2FF	0101	000 0000	111 1111
PLASTIC	**6**	**0 - 111**	**768 - 879**	**300 - 36F**	**0110**	**000 0000**	**110 1111**
PLASTIC Space Weather	**6**	**112 - 127**	**880 - 895**	**370 - 37F**	**0110**	**111 0000**	**111 1111**
Reserved	7	0 - 127	896 - 1023	380 - 3FF	0111	000 0000	111 1111
SECCHI	**8**	**0 - 111**	**1024 - 1135**	**400 - 46F**	**1000**	**000 0000**	**110 1111**
SECCHI Space Weather	**8**	**112 - 127**	**1136 - 1151**	**470 - 47F**	**1000**	**111 0000**	**111 1111**
Reserved	9	0 - 127	1152 - 1279	480 - 4FF	1001	000 0000	111 1111
SWAVES	**10**	**0 - 111**	**1280 - 1391**	**500 - 56F**	**1010**	**000 0000**	**110 1111**
SWAVES Space Weather	**10**	**112 - 127**	**1392 - 1407**	**570 - 57F**	**1010**	**111 0000**	**111 1111**
Reserved	11	0 - 127	1408 - 1535	580 - 5FF	1011	000 0000	111 1111
MOC	12	0 - 127	1536 - 1663	600 - 67F	1100	000 0000	111 1111
Spacecraft Subsystem	13	0 - 127	1664 - 1791	680 - 6FF	1101	000 0000	111 1111
GSE	14	0 - 127	1792 - 1919	700 - 77F	1110	000 0000	111 1111
Unassigned	15	0 - 127	1920 - 2047	780 - 7FF	1111	000 0000	111 1111

Note: The 11 bit CCSDS ApID is sub-divided by the STEREO project such that:
The 4 Most Significant Bits are used to identify the relevant Subsystem
The 7 Least Significant Bits are used to identify the Data Format used in the packet

APPENDIX C. TELEMETRY INTERFACE USER GUIDE EXCERPT

This section is aimed at the end user who has built a client interface to the Telemetry Server Socket Interface. A brief description of all the functionality that the user may interact with is given. Common errors are addressed.

A POC user chooses real-time telemetry delivery service or archive playback service by the choice of host and port he or she connects to. The real-time telemetry process (router) listens on a specific port and the archive playback process (archive_server) listens on a different port. After making a connection, the client sends a series of ASCII directive messages that identify the characteristics of the desired telemetry and then request the server to begin sending the telemetry.

C.1 REQUESTING TELEMETRY

INPUT

The user specifies which data are desired by sending directives to the telemetry server over a socket connection. The telemetry server expects to receive directives in uppercase ASCII format.

OUTPUT

The POC user can select data to be returned in one of two formats: Supplemented Telemetry Packet (STP), or Payload Telemetry Packet (PTP).

The telemetry server sends any data that match the user's request in the form requested. The data are in binary representation in network byte order.

If a user connects to the router and there is no telemetry matching the user's specification, the router will maintain the connection but will flow no telemetry. If a user connects to the archive_server and there is no telemetry matching the user's specification, the archive_server will maintain the connection but will send a GRH with the type field set to "EOT". It is expected that the client will close the connection upon receipt of an EOT.

CLOSING THE CONNECTION

After a request has been received and the appropriate data sent (in some cases no data), the telemetry server does not close the socket connection. The user/tool must close the connection.

A client should not expect to receive any confirmation of a connection or receipt of directives.

NOTE: For the examples below, telnet is used and the binary information that is sent from the Telemetry Server will be unreadable. Telnet is used here as a simple way to illustrate the process of connecting to an output port. Telnet can be used to quickly confirm that the desired information exists and directives are sent correctly.

A fairly sophisticated example of a client tool to request telemetry will be provided separately from this document.

C.2 REAL-TIME TELEMETRY

It is possible to receive real-time telemetry from the router as it is being received. To do this, simply connect to the telemetry server's host machine on the appropriate port (output port for a router). If you are successfully connected, nothing will appear; the application is waiting for you to specify the type of data to output. Note that the telemetry specification directives shown below are all optional; only the BEGN RT is required. The values shown below as arguments to the directives are the actual default values used if you don't specify the directive. These directives are included to demonstrate correct syntax.

EXAMPLE

```
>telnet [hostname] 3102
>SSYS ALL
>SRCE ALL
>STRM ALL
>FRNT ALL
>PATH ALL
>VCHN ALL
>TYPE STP
>BEGN RT
```

Once the BEGN RT directive has been sent, no more directives or changes to directives will be read in. The connection must be closed and reopened to receive different data.

END OF DATA

The router cannot indicate to the output client that the end of data has been reached since it is simply passing data along as it receives it.

C.3 PLAYBACK TELEMETRY

The archive_server process plays back archived telemetry. To access archived telemetry, simply telnet to the host machine on the specified port. If you are successfully connected, nothing will appear; the application is waiting for you to specify the type of data to output. Note that the telemetry specification directives shown below are all optional; only the BEGN PB is required. The values shown below as arguments to the directives are the actual default values used if you don't specify the directive (except for the STRT / STOP directives, which are effectively set to the beginning and end of the current day if they are not explicitly specified). These directives are included to demonstrate correct syntax. The following is an example of what to type:

```
>telnet [hostname] 3302
>SSYS ALL
>SRCE ALL
>STRM ALL
>FRNT ALL
>PATH ALL
>VCHN ALL
>TYPE STP
>STRT 2003 300 01:00:00
>STOP 2003 300 23:00:00
>BEGN PB
```

Once the directives have been received and a query started, the archive_server will not accept any more directives.

C.3.1 TELEMETRY TIME ORDER DURING PLAYBACK

Unlike the real-time connection, a time range can be specified when playing back telemetry from the archive. Data can be requested in ground receipt time (GRT) order or spacecraft time (SCT) order. The STRT and STOP directives are used to specify the desired range. If a STRT directive is not given, playback will begin with telemetry from the beginning of the current day. If a STOP directive is not given, playback will terminate with the latest data in the archive.

The GRT is read in from the Ground Receipt Header and is in a DSN Time Format for STEREO (Days since Jan 1, 1958 UTC and Milliseconds of day). The format for requesting time for the STRT or STOP directive is the following:

YYYY DDD HH:MM:SS - year, day of year, hour, minute, second

The SCT is read in from the Secondary Packet Header and is in STEREO-UTC Time for STEREO. SCT can be specified in two formats, either the Unix Time equivalent in GRT (in this case the user-entered Ground Time will be mapped to a spacecraft time) or by the partition number/STEREO-UTC counts (in this case the user must enter the actual value that was found in the Spacecraft time register). The format for requesting time for the STRT or STOP directive is the following:

YYYY DDD HH:MM:SS - year, day of year, hour, minute, second
Partition Number/ UTC-STEREO counts in seconds

C.3.2 END OF DATA

A Ground Receipt Header with the data type set to zero will be sent to indicate that the data transfer is complete. This is an EOT (End of Transmission) Data Type. The archive_server determines if the data transmission is finished by checking the value of the STOP time directive with the data being sent.

C.3.3 CLOSING THE CONNECTION

When all the data has been sent, or if no data match the clients request, the archive_server will send an EOT (End of Transmission) message type. The archive_server does not close the socket connection in either case. The user/tool must close the connection.

C.4 DIRECTIVES

The following are syntax rules to be aware of when sending directives.

1. Directives must be ASCII characters
2. Directives and Parameters must be in uppercase letters
3. It does not matter whether a space is placed between a directive and parameter
4. Directives should be sent one at a time (i.e., delimited by newlines)

Following are some additional notes regarding the behavior of the server in response to particular combinations of APID-related directives:

1. Directives: No SSYS specified; No APID specified; No EXAPID specified
   Result: client will receive every APID available

2. Directives: No SSYS specified; No APID specified; EXAPID specified
   Result: client will receive every APID available except those specified by EXAPID

3. Directives: No SSYS specified; APID(s) specified; EXAPID is irrelevant in this case
   Result: client will receive only the APIDs specified by APID(s)

4. Directives: SSYS(s) specified; No APID specified; No EXAPID specified
   Result: client will receive every APID available within the SSYS subsystems

5. Directives: SSYS(s) specified; No APID specified; EXAPID specified
   Result: client will receive every APID available within the SSYS subsystems except those specified by EXAPID(s)

6. Directives: SSYS(s) specified; APID(s) specified; No EXAPID(s) specified
   Result: client will receive every APID available within the SSYS subsystems and the APIDs specified by APID(s)

7. Directives: SSYS(s) specified; APID(s) specified; EXAPID(s) specified
   Result: client will receive every APID available within the SSYS subsystems except those specified by EXAPID(s), and the APIDs specified by APID(s)

APPENDIX D. ACRONYMS AND ABBREVIATIONS

AH	Attitude History
AOS	Acquisition of Signal
APID	Application Identifier
APL	Applied Physics Laboratory
APL	Archived Product List
ARR	Authorization Return Receipt
ASCII	American Standard Code for Information Interchange
ATP	As-run Track Plan
BPS	Bits Per Second
C&DH	Command and Data Handling
CCSDS	Consultative Committee for Space Data Systems
CH	Command History
CMD	Command
COTS	Commercial Off the Shelf System
CSH	Converted Spacecraft Housekeeping
DLL	Data Loss Log
DP	Momentum Dump Prediction
DPU	Data Processing Unit
DS	DSN Schedule
DSMS	Deep Space Mission System
DSN	Deep Space Network
DT	Daily Timeline
EPM	Ephemerides
ERT	Earth Receipt Time
FDF	Flight Dynamics Facility
Gbits	Giga bits
G&C	Guidance and Control
GT	SECCHI Guide Telescope Filtered Data
HGA	HGA Gimbal Angles
HK	Housekeeping
I&T	Integration and Test
IDPU	Instrument Data Processing Unit
IES	Instrument Event Schedule
I/F	Interface
IP	Internet Protocol
JHU/APL	The Johns Hopkins University/Applied Physics Laboratory
JPL	Jet Propulsion Laboratory
KB, KByte	KiloByte
Kbps	Kilobits Per Second
Mb	Megabit
MB, Mbyte	Megabyte
MET	Mission Elapsed Time
MOC	Mission Operations Center
MOT	Mission Operations Team
MP	Maneuver Plan
MRC	Maneuver Radial Component
MSR	MOC Status Report
N/A	Not Applicable
NASA	National Aeronautics and Space Agency
OAR	Out of Archive Time Range
PH	Telecommand Packet History
PLASTIC	Plasma and Suprathermal Ion Composition
PDS	Post-momentum Dump Summary
PDT	Post-momentum Dump Thrust History
PMR	Post Maneuver Report
POC	Payload Operations Center

PPP	Point to Point Protocol
PTP	Payload Telemetry Packet
S/C	Spacecraft
SCB	Stored Command Buffer
SECCHI	Sun Earth Connection Coronal and Heliospheric Investigation
SFDU	Standard Formatted Data Unit
SOE	Sequence of Events
SSC	STEREO Science Center
STEREO	Solar-TErrestrial RElations Observatory
STP	Supplemented Telemetry Packet
SV	State Vectors
SWAVES	STEREO WAVES
TBD	To Be Determined
TCP/IP	Transmission Control Protocol/Internet Protocol
TD	Telemetry Dictionary
TH	Time History
TLM	Telemetry
TP	Track Plan
TTF	Telemetry Transfer Frame
URL	Uniform Resource Locator
UT	Universal Time
UTC	Universal Time Coordinated
VC0	Virtual Channel 0 (Fill data)
VC6	Virtual Channel 6 (SSR dump)
VC7	Virtual Channel 7 (real-time)
VPD	DSN Viewperiod
VPF	FDF Viewperiod
WS	Weekly Schedule
XSP	Ephemerides to DSN